MONSOON
PAN Run-time Configuration, Setup and Operating Mode Definition Interface Description

NOAO Document ICD 5.1
Revision: 1.0

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

Authored by:
Nick C. Buchholz and Phil N. Daly
12/4/2003
Please send comments:
nbuchholz@noao.edu

Doc. File: ICD_5.1 PAN Config_v1_0.doc
Doc. Number MNSN-AD-01-0007
Created on 8/17/2006
Page 1 of 27
## Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date Approved</th>
<th>Sections Affected</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1.1</td>
<td>12/4/2003</td>
<td>All</td>
<td>Second release draft</td>
</tr>
<tr>
<td>1.0 (Provisional)</td>
<td>8/17/2006</td>
<td>All</td>
<td>Reformatted and edited - aro</td>
</tr>
</tbody>
</table>
Table of Contents

Revision History ................................................................................................................. 2
Table of Contents ............................................................................................................... 3
List of Figures .................................................................................................................... 4
List of Tables .................................................................................................................... 4
1.0 Introduction ............................................................................................................... 5
   1.1 Scope ..................................................................................................................... 5
   1.2 Purpose .................................................................................................................. 6
   1.3 Acronyms and Glossary ......................................................................................... 6
   1.4 Standard Terminology ......................................................................................... 10
   1.5 What’s in This Document ...................................................................................... 11
   1.6 What’s NOT in This Document ............................................................................. 11
2.0 PAN Software Setup ................................................................................................... 12
3.0 Configuration Techniques, General Requirements and Philosophy ....................... 12
   3.1 Shared Libraries – System Library Changes ..................................................... 13
   3.2 Detector Specific Commands and Methods ....................................................... 13
   3.3 Configuration Files - .arr, .cfg, .gui ................................................................. 13
4.0 Initialization, Setup and observation/Exposure Modes ............................................. 14
   4.1 Initialization Record - .ini .................................................................................... 14
   4.2 Setup Records - .mod, .ucd, .txt ....................................................................... 14
5.0 The Detector Command Library - libDetCmnds ...................................................... 14
   5.1 libDetCmnds – Required Functions .................................................................... 14
6.0 The Detector/Focal Plane Descriptor Record – xxx.arr ......................................... 18
7.0 The PAN/DHE Attribute Description Record – xxx.csv ....................................... 18
8.0 The GUI Interface Configuration Record – xxx.guiCategories.txt ..................... 18
9.0 The Initialization Record – xxx.ini ......................................................................... 18
10.0 The Setup Routines – xxx_DefaultSetup.mod ..................................................... 18
11.0 Observation and Exposure Modes –
       xxx_Exp.mod, xxx_Idp.mod, xxx_SeqV00#.ucd .................................................. 18
Appendix I Commands and Defined Variables and Parameters .................................... 19
Appendix II Attribute-Value Pair Notation Conventions ............................................. 23
Appendix III Configuration File Format ......................................................................... 24
Appendix IV Memory Configuration File Format ......................................................... 26
List of Figures

Figure 1 – Observatory Reference Model One .............................................................5
Figure 2 – MONSOON Context Diagram.................................................................12

List of Tables

Table 1 – Commands and Defined Variable and Parameters......................................19
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 which configure the detector, generate and preprocess the data, and send the data to an archive for later analysis.

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

One of the basic requirements placed on the MONSOON project software was that the resulting programs should be configurable at run-time to appropriately describe and control any focal plane controlled by a MONSOON image server system.

This document delineates the interface, formats and methods used to do this run-time configuration and the formats and methods required to initialize and setup operating modes in the PAN/DHE software.

---

**Observatory System Reference Model One**

![Diagram of Observatory System Reference Model One](image)

**Observatory System Reference Model One**

Figure 1

---

Doc. File: ICD_5.1 PAN Config_v1_0.doc

Doc. Number MNSN-AD-01-0007

Created on 8/17/2006

Page 5 of 27
1.2 Purpose

This Interface Control Document (ICD) serves four purposes:

- To describe the run-time configuration system of the MONSOON PAN software.
- To describe the method used to initialize the parameters required by the software.
- To describe the method used to define operation Modes for a particular focal plane.
- To describe the behaviour of the run-time configuration system and the initialization system used by the PAN software.

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.

1.3 Acronyms and Glossary

1.3.1 Abbreviations and Acronyms

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

**Attribute**
An entity that describes some aspect of the configuration of a Detector Head Electronics, Image Acquisition System or science instrument. Examples are: the name of a filter, a DAC voltage value, or the tilt angle of a grating. Some attributes will be used by the Instrument Control System as command parameters. The OCS communicates with a science instrument by sending it sets of attributes and values.

**Byte**
8 bits.

**Command**
An instruction requiring a system to start some action. The action may result in a voltage changing or some internal parameters being set to particular values. A command may have command parameters (arguments) that contain the details of the instruction to be obeyed.

**Data Array**
The data, while it is stored in data processing memory, which resulted from one or more readouts of an IR array or CCD detector.

**Data Set**
A self-contained collection of data generated as a result of a Pixel Server obeying a `gpxStartExp` command. Each `gpxStartExp` command results in one and only one data set.

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

**Exposure**
The process and the data resulting from the process of resetting or clearing a detector, exposing it to photons and then reading one or more frames to determine the photon levels. These frames are processed into a data array, called an exposure, which may be further processed. For example, an exposure would be the data array that results when a single Reset-Readout-Integrate-Readout cycle is performed on an IR detector or a single CCD Clear-Integrate Readout cycle.

**Frame**
The result of a single readout of an array. Each frame represents the signal values obtained from reading the entire ROI being read out of the detector. Multiple frames may be processed into a single exposure.
1.3.2 Glossary (Cont.)

**Generic Pixel Server (GPX)**  
A set of software routines that implement a pixel server. A Generic Pixel Server does not require any particular set of proprietary hardware or software.

**Image**  
The array of detector pixel and description data representing a science or diagnostic exposure or combined set of exposures. An image is capable of being displayed or processed as a discrete entity. The values in the array may be stored in memory or on disk and are related to the data taken by the detector by some processing algorithm. For example, an image may consist of all the co-added and averaged exposures in one beam of a chop mode `gpxStartExp` command.

**Image Acquisition System**  
A system of software and hardware capable of producing images from a detector on command.

**Image Server**  
A system of software and hardware capable of producing images from a detector on command.

**Instrument Control System**  
A set of software routines designed to control and configure a science instrument to take science observations.

**MONSOON Image Acquisition System**  
A Generic Pixel Server. An extensible, modular Image Acquisition System. The design of the system is, to the extent possible, independent of the hardware being used in a particular implementation. Each component of the system should be capable of replacement by a similar component without having to redesign the rest of the system. Each component of the software is, as far as possible, independent of the underlying hardware and as modular as possible.

**Observation**  
The process of exposing the detector to photons through the telescope in one or more exposures. The result of an observation is an image.

**Pixel Acquisition Node**  
The computer that handles the interface to the Detector Head Electronics and the image pre-processing of the data stream from the Detector Head Electronics.

**Pixel Server**  
A set of software and hardware that accepts commands and produces a set of pixels (an image) related to those commands.

**Read**  
When used as a noun to describe instrument data, a single read of a pixel on the detector. A read may consist of several A/D conversions of the pixel data that are averaged or processed in some other way to produce a single integer output value for the pixel. A readout is made up of one read of each pixel in the detector ROI being read.
1.3.2 Glossary (Cont.)

**Readout**
When used as a noun to describe instrument data, a single read of every pixel on the detector. A frame is made up of one or more readouts averaged pixel by pixel.

**Region of Interest (ROI)**
A sub-array of the available detector area. There are two types of sub-arrays that can be defined. The Sequence ROI is an ROI on the active surface of the array used to increase the frequency of the array readout. The Data Reduction ROI is an arbitrary rectangle of any size that fits on the array. Data Reduction ROIs are defined to reduce the volume of data sent to the disk or DHS even when the entire array is being read out.

**Value**
The value associated with an attribute.

**Word**
Four bytes or 32 bits.

1.3.3 Reference Documents

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

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

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

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


ANSI/IEEE Std 754-1985 - “IEEE Standard for Binary Floating-Point Arithmetic” - Standards Committee of the IEEE Computer Society, USA 19850812

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


NOAO Document MNSN-AD-01-0003 - ICD 5.0 Version 1.0 - PAN Communications, Command/Response and Data Stream Interface Description, Nick C. Buchholz (NOAO), Phil N.Daly (NOAO), 8/15/2006

NOAO Document MNSN-AD-01-0004 - ICD 6.0 Version 1.0 - Generic Detector Head Electronics Command and Data Stream Interface Description, Nick C. Buchholz (NOAO), Barry M. Starr (NOAO), 8/9/2006

1.4 Standard Terminology

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

- RULE
- RECOMMENDATION
- SUGGESTION
- PERMISSION
- OBSERVATION

These keywords are used as follows:

**RULE**

<Paragraph Number> Subject Describing Text

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:

2.5.1 GPX Names

**SUGGESTION**

<Paragraph Number> Subject Describing Text

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

Example:

2.5.2 Long Variables Names
PERMISSION

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

Example:
2.6 Long Variables Names

OBSERVATION

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

Example:
2.7 Long Variables Names

1.5 What’s in this Document

This document presents the methods and procedures used to configure, initialize and control the PAN command set and attributes. It includes the requirements for configuration records and a description of the interface. It also includes a procedure which can be used to create a new configuration file, Initialization and setup file set. Finally, the document describes the procedure for describing and developing Observing/exposure modes for use with new focal planes.

1.6 What’s NOT in this Document

This document does not describe the internal command and communications protocols or the nature of the internal command passing techniques. In particular, the hardware interconnects and data passing techniques within a PAN software are explicitly excluded from this discussion.
2.0 PAN Software Startup

The MONSOON PAN software is started by running the script panStartup.csh on the PAN Linux box. This script should only be needed once unless there is a power glitch, or a ppxExit command is issued by the Client system. The panStartup.csh script starts up the pan processes, (panDaemon, panCapture, panProcAlg and panSaver). These processes then control the initialization, setup and use of the PAN/DHE system and allow the capture pre-processing and archiving of image data from the focal plane.

3.0 Configuration Techniques, General Requirements and Philosophy

The MONSOON PAN software requirements document demands that the MONSOON PAN software be re-configurable to accommodate new array types and new focal planes without recompiling the entire software system. By describing a number of interfaces and API’s between various parts of the system this requirement has been met. Several levels of re-configuration are provided as briefly described in the following paragraphs. Currently the configuration descriptions are kept in text files. Future descriptions may be stored in database records and or XML files. The software should continue to support all three styles of configuration record.
3.1 Shared Libraries - System Library Changes

The first, most comprehensive and least recommended configuration method is to rewrite the MONSOON system shared libraries. When the PAN software starts up, part of the load process includes the loading of the shared libraries which implement the basic routines used in the software. One method of re-configuring the software would be to modify these basic shared libraries. **This is not recommended.** The MONSOON shared libraries are being and have been thoroughly tested to insure they meet the API for the routines and work correctly within the MONSOON software. This testing is ongoing. While we try to avoid the use of side effects in routines, we can’t guarantee that changes will not have unintended consequences.

3.2 Detector Specific Commands and Methods - libdetCmnds

The next most flexible configuration method is the use of the detector command library. Each detector type and focal plane configuration can have a set of commands to do the unique basic tasks required in controlling the detector or focal plane. This library can adopt commands from the generic command library and modify only those commands which are unique to the focal plane. A complete description of the structure, contents and use of the libdetCmnds library is included in a later section.

3.3 Configuration Files - .arr, .cfg, .gui

Currently the software loads in three configuration records during the start-up process. These records, with the libdetCmnds library, are sufficient to configure the MONSOON PAN software to run most array and focal plane configurations. The details of these record formats and contents are described in later major section.

3.3.1 Focal Plane Description Record - the .arr file

This record contains a description of the geometry of the focal plane and the detectors which make it up. The information in this record will determine the size of buffers, the number of output channels in the focal plane and the general structure and layout of the focal plane.

3.3.2 PAN/DHE Attribute Description Record - the .cfg file

This record is partially created by the Electronic or Detector Engineer doing the integration of the MONSOON DHE hardware and the detector Focal Plane and partially by the software integrator. It contains a table of values which describe all of the settable attributes in the PAN/DHE system and gives meaningful names to DHE attribute functions; such as SEQ_LOOP_REG[1] in the DHE might be defined as rows2read in the PAN and clients.
3.3.3 The GUI Interface Configuration Record - the .gui file

This record contains a list of category names which describe groups of attributes to be provided by the engineering or science client GUI. These groups are determined by the hardware integrator and or the Science instrument team.

4.0 Initialization, Setup and Observation/Exposure Modes

Once the software is running and properly configured the Client program can initialize the system, set up the detector/focal plane parameters and set up observation and exposure modes for the focal plane and detectors. To do this the client system sends the GPX the gpx[ppx]SetMode or gpx[ppx]SetSysCfg commands. These commands use the record name on the command line or the record name set into the appropriate PAN attribute to initialize or setup the PAN-DHE pair.

The set mode commands may be used multiple times to initialize the setup or to refine a previously loaded mode.

4.1 Initialization Record - .ini

4.2 Setup Records - .mod, .ucd, .txt

5.0 The Detector Command Library -libDetCmnds

The detector command library is a library built for each focal plane which encapsulates the unique details required to run the focal plane in question. The library contains a set of commands which must appear in the library and my contain command names which are unique to a particular detector or focal plane. The standard commands are explained below. Each detector specific library MUST include all of the commands listed in the required commands section and may use the generic versions or focal plane specific versions of the commands.

5.1 libDetCmnds - Required Functions

5.1.1 Hardware Attribute Commands

These commands are the commands used to set and read back attributes in the DHE and PAN hardware. The current set of methods is compiled into the detector specific library. If no new methods are created, then only the detector library need be changed to change the set readback behavior of the PAN/DHE. If new commands are added (recommended only when required), then the additional commands will require changes to the cmdCfgUtil library code to reflect these additional commands. Note that this should only be required in cases where a significant change is instituted in the underlying MONSOON DHE hardware.
5.1.1.1 setBiasV, getBiasV
void setBiasV ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getBiasV ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.2 setFBias, getFBias
void setFBias ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getFBias ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.3 setImageName, getImageName
void setImageName ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getImageName ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.4 setIntTime, getIntTime
void setIntTime ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getIntTime ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.5 noMethod
void noMethod ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.6 setRdMskWrt, getRdMskWrt
void getRdMskWrt ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void setRdMskWrt ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

void _dheSetRdMskWrt(long *istat, char *resp, cmdTblP attAddr, char *attValue);
void _dheGetRdMskWrt(long *istat, char *resp, cmdTblP attAddr, char *attValue);

5.1.1.7 setRoiCols, getRoiCols
void setRoiCols ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getRoiCols ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.8 setRoiRows, getRoiRows
void setRoiRows ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getRoiRows ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.9 setSimple, getSimple, _panSetSimple, _dheSetSimple, _panGetSimple, _dheGetSimple
void getSimple ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void setSimple ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void _panSetSimple(long *istat, char *resp, cmdTblP attAddr, char *attValue);
void _dheSetSimple(long *istat, char *resp, cmdTblP attAddr, char *attValue);
void _panGetSimple(long *istat, char *resp, cmdTblP attAddr, char *attValue);
void _dheGetSimple(long *istat, char *resp, cmdTblP attAddr, char *attValue);
5.1.1.10  setString, getString
void setString ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getString ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.1.11  setVOffset, getVOffset
void setVOffset ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void getVOffset ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);
void _dheSetVOffset(long *istat, char *resp, cmdTblP attAddr, char *attValue);

5.1.1.12  wrtToRead
void wrtToRead ( long *istat, char *resp, ulong func, cmdTblP attAddr, char *attValue, ulong dheState);

5.1.2  Detector Specific Function Commands

5.1.2.1  detInit
5.1.2.2  detUninit
5.1.2.3  detAbortExp
void detAbortExp    ( long *istat, char *resp );

5.1.2.4  detArmExpTrigger
void detArmExpTrigger    ( long *istat, char *resp );

5.1.2.5  detAsyncResp
void detAsyncResp    ( long *istat, char *resp, cmdTblP baddr);

5.1.2.6  detGetState
void detGetState    ( long *istat, char *resp, void *sPtr);

5.1.2.7  detPauseExp
void detPauseExp    ( long *istat, char *resp);

5.1.2.8  detPower
void detPower    ( long *istat, char *resp, ulong state, ulong sequence);

5.1.2.9  detReset
void detReset    ( long *istat, char *resp, ulong boards, ulong level);

5.1.2.10  detResumeExp
void detResumeExp    ( long *istat, char *resp);

5.1.2.11  detShftImg
void detShftImg    ( long *istat, char *resp, ulong shftX, ulong shftY);

5.1.2.12  detShutDown
void detShutDown    ( long *istat, char *resp);

5.1.2.13  detShutter
void detShutter    ( long *istat, char *resp);

5.1.2.14  detSimulate
void detSimulate    ( long *istat, char *resp, ulong system, ulong level);
5.1.2.15 detStartExp
void detStartExp ( long *istat, char *resp);

5.1.2.16 detStopExp
void detStopExp ( long *istat, char *resp);

5.1.2.17 detTestMode
void detTestMode ( long *istat, char *resp, ulong testSuite, ulong level, ulong *result);

5.1.3 Detector Specific Data Handling Commands

5.1.3.1 detDescramble
void detDescramble ( long *istat, char *resp, long *src, int sPxlSize, long *dest, int dPxlSize);

5.1.3.2 detProcess
void detProcess ( long *istat, char *resp, long *dest);

5.1.3.3 imgAdd
void imgAdd (long *istat, char *resp, void *srcBuf, int sPxlSize, void *destBuf, int dPxlSize, long numPix);

5.1.3.4 imgSub
void imgSub (long *istat, char *resp, void *srcBuf, int sPxlSize, void *destBuf, int dPxlSize, long numPix);

5.1.3.5 imgMov
void imgMov (long *istat, char *resp, void *srcBuf, int sPxlSize, void *destBuf, int dPxlSize, long numPix);

5.1.3.6 detCalcPixels
void detCalcPixels ( long *istat, char *resp, long *numPixels);

5.1.3.7 detCapture
void detCapture ( long *istat, char *resp);

5.1.4 Internal Helper Commands - Provided in _generic

5.1.4.1 _cnvrtToDHE
void _cnvrtToDHE (long *istat, char *resp, cmdTblP attAddr, double fValue, ulong *dValue);

5.1.4.2 _cnvrtToPAN
void _cnvrtToPAN (long *istat, char *resp, cmdTblP attAddr, ulong dValue, double *fValue);

5.1.4.3 _chkRange
void _chkRange (long *istat, char *resp, cmdTblP attAddr, char *attValue);

5.1.4.4 _chkValue
void _chkValue (long *istat, char *resp, cmdTblP attAddr, double fValue);
6.0 The Detector/Focal Plane Descriptor Record - xxx.arr

7.0 The PAN/DHE Attribute Description Record - xxx.csv

8.0 The GUI Interface Configuration Record - xxx_guiCategories.txt

9.0 The Initialization Record - xxx.ini

10.0 The Setup Routines - xxx_DefaultSetup.mod

11.0 Observation & Exposure Modes - xxx_Exp.mod, xxx_Idp.mod, xxx_SeqV00#.ucd
### Appendix I

#### Table 1 – 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>ppxSetMode</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 a 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 which outlines how the array outputs are read. This includes a queue descriptor which tells where to place the pixels for processing and information on the structure of the pixel data block transferred. We describe the block of data as chunks of contiguous pixels separated by a intervening pixels from other blocks. The block is described by a number of integers giving the starting row and column of the block, a row and column stride (the number of pixels to skip when storing chunks) the row and column chunk size and the total size of the final block of data in rows and columns.</td>
<td></td>
</tr>
<tr>
<td>spcDescriptor</td>
<td>A structure which describes the configuration of the signal processor chains in the system. The components of the structure are floating point arrays which describe the gain, settling time, and offset of the signal processing chain. Included is a noise figure of merit (TBD) which will allow the quietest set of chains to be chosen when that is important.</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 which 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 which are to be loaded into any DAC settable voltages used to control the array. Each system will likely have a unique set of these voltages and a mapping from voltage name to DAC number should be provided in the P PX.</td>
<td></td>
</tr>
<tr>
<td>Min Integration</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>
</tbody>
</table>
### Table 1 – Commands, Defined Variables and Parameters (Cont.)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Parameters Set/Controlled</th>
<th>Usage/Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppxSetArrConfig (Cont.)</td>
<td>spdRoiDescriptor Row0 Col0 rowSize ColSize</td>
<td>A structure which describes a “speed up” region of interest (ROI). This is provided so a system with a large array can describe a sub array which will be readout to provide faster readout and shorter integration times. (Mostly used for IR systems without an internal cold shutter) The ROI is described by four integers giving the First row and column to be read and the size of the ROI in rows and columns.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>binning</td>
<td></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>
</tr>
<tr>
<td>intTimeSecs</td>
<td></td>
<td>A floating point number giving the desired integration time to use in seconds.</td>
</tr>
<tr>
<td>digAvgs</td>
<td></td>
<td>An integer giving the Number of Digital Averages to use while reading out the pixels.</td>
</tr>
<tr>
<td>numPics</td>
<td></td>
<td>An integer giving the Number of pictures to generate for each PXPStartExp.</td>
</tr>
<tr>
<td>ROIdescrptors Row0 Col0 rowSize ColSize</td>
<td></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>
</tr>
<tr>
<td>Outputs to Read</td>
<td></td>
<td>An integer giving the number of outputs on the array which will be used during the readout.</td>
</tr>
<tr>
<td>PreFlash</td>
<td></td>
<td>A Boolean value determining if the exposure sequence will include a pre-flash step.</td>
</tr>
<tr>
<td>waveFormsToRun</td>
<td></td>
<td>A list of the waveforms to run during this array readout.</td>
</tr>
<tr>
<td>shutterState</td>
<td></td>
<td>A Boolean value determining if the shutter is to be opened during the integration time.</td>
</tr>
<tr>
<td>arrayPowerState</td>
<td></td>
<td>A Boolean value determining if the array will be activated/powered up during the exposure.</td>
</tr>
<tr>
<td>intraPixelDelay</td>
<td></td>
<td>A floating point number giving the amount of time to allow for settling while reading each pixel.</td>
</tr>
<tr>
<td>idleProcess</td>
<td></td>
<td>An integer tag describing how the array will be run during any idle time in the observing run.</td>
</tr>
</tbody>
</table>
### Table 1 – Commands, Defined Variables and Parameters (Cont.)

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Parameters Set/Controlled</th>
<th>Usage/Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppxSetIDPConfig</td>
<td>Data Disposition - struct&lt;br&gt;disposition - procedure name&lt;br&gt;Arguments - filename, directory&lt;br&gt;image format, data type, data stream/queue&lt;br&gt;Pre-processing Algorithm&lt;br&gt;Unscrambling Algorithm&lt;br&gt;Image Data Set ID&lt;br&gt;coAdds - integer&lt;br&gt;fSamples - integer</td>
<td></td>
</tr>
<tr>
<td>ppxSetAVP</td>
<td>Every user-settable attribute</td>
<td></td>
</tr>
<tr>
<td>ppxStartExp</td>
<td>binning - Integer&lt;br&gt;intTimeSecs – float&lt;br&gt;digAvgs - integer&lt;br&gt;numPics – integer&lt;br&gt;shutterState&lt;br&gt;arrayPowerState&lt;br&gt;Data Disposition&lt;br&gt;Pre-processing Algorithm&lt;br&gt;Unscrambling Algorithm&lt;br&gt;Image Data Set ID&lt;br&gt;coAdds – integer&lt;br&gt;fSamples - integer</td>
<td></td>
</tr>
<tr>
<td>ppxArmExpTrigger</td>
<td>TriggerSource&lt;br&gt;TriggerTime Out</td>
<td></td>
</tr>
<tr>
<td>ppxStop</td>
<td>Image Data Set ID</td>
<td></td>
</tr>
<tr>
<td>ppxAbort</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>ppxPause</td>
<td>intTimeSecs – float</td>
<td></td>
</tr>
<tr>
<td>ppxResume</td>
<td>intTimeSecs – float</td>
<td></td>
</tr>
<tr>
<td>ppxShutter</td>
<td>Current shutter state</td>
<td></td>
</tr>
<tr>
<td>ppxShft/Image</td>
<td>Row or Y shift&lt;br&gt;Column or X shift</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>ppxSimulate</td>
<td>Units to simulate</td>
<td></td>
</tr>
<tr>
<td>ppxTestMode</td>
<td>Units to test</td>
<td></td>
</tr>
<tr>
<td>ppxPower</td>
<td>System Power State</td>
<td></td>
</tr>
<tr>
<td>ppxReset</td>
<td>System Reset Level</td>
<td></td>
</tr>
<tr>
<td>ppxGetState</td>
<td>Reads all system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>state attributes</td>
<td></td>
</tr>
<tr>
<td>ppxGetAVP</td>
<td>Reads every</td>
<td></td>
</tr>
<tr>
<td></td>
<td>individual system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>attribute</td>
<td></td>
</tr>
<tr>
<td>ppxAsynchMsg</td>
<td>Reports asynchronous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>events and errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Asynchronously</td>
<td></td>
</tr>
<tr>
<td></td>
<td>reports attribute</td>
<td></td>
</tr>
<tr>
<td></td>
<td>values</td>
<td></td>
</tr>
<tr>
<td>ppxRstAsynStatus</td>
<td>Resets asynchronous</td>
<td></td>
</tr>
<tr>
<td></td>
<td>event and error</td>
<td></td>
</tr>
<tr>
<td></td>
<td>flags</td>
<td></td>
</tr>
<tr>
<td>ppxPass</td>
<td>Passes string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>command to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>underlying system</td>
<td></td>
</tr>
</tbody>
</table>
Appendix II – 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 III – Configuration File Format

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

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

  sectionName :: | { GENERAL | ARRAY_VOLTAGES | ARRAYCLOCKS | VIDEOCHANNELS | READOUTPARAMS | EXPOSUREPARAMS | DATAPREPROCESSING } } \n
- **Items in Bold** indicate that the exact bold text is to appear in the configuration file. \n in a description indicates the presence of a newline character or CR-LF sequence in the file.

- **Items in Italic** 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 :: *modelIDLLine* [ *section* ]^+

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

- Items enclosed in curly braces and separated by vertical bars i.e. \{ A | B \} indicate a choice. Thus

  [ \{TRUE | FALSE\} ] indicates that either TRUE or FALSE would appear between the square braces.

- Items in square braces i.e. \[ A \] indicate optional items that may appear in the file or may be omitted.

- Items in square braces with a asterisk or plus sign super script indicate 0 (*) or 1(+) or more of the optional items. Thus \[ Attribute-Value Pair \]^+ indicates the presence of one or more attribute value pairs.
The description of the Configuration File follows:

configurationFile :: modeIdLine fileSections

modeIdLine :: ModeName = FileName 

fileSections :: [ generalSection array ] [ VoltageSection ] [ arrayClockSection ]
[ videoChannelSection ] [ readoutParamsSection ]
[ exposureParamSection ] [ dataPreProcessSection ]

generalSection :: [GENERAL] 

sectionLines

VoltageSection :: [ARRAYVOLTAGES] 

sectionLines

arrayClockSection :: [ARRAYCLOCKS] 

sectionLines

videoChannelSection :: [VIDEOCHANNELS] 

sectionLines

readoutParamSection :: [READOUTPARAMS] 

sectionLines

exposureParamSection:: [EXPOSUREPARAMS] 

sectionLines

dataPreProcessSection:: [DATAPREPROCESSING] 

sectionLines

sectionLines :: [ sectionConfigurationLines ] * [AttributeValueLines ] *
Appendix IV – Memory Configuration File Format

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

The description of the download file format follows the conventions described above in Appendix III for configuration files.

The description of the Download File format follows:

```plaintext
configurationFile :: dwnLdLinesList
dwnLdLinesList :: fileLine [ dwnLdLinesList ]
fileLine :: { commentLine | addressLine | valueLine | blankLine }
commentLine :: # commentString \n
addressLine :: addr= { attributeName | absoluteAddress } [ # commentString ] \n
valueLine :: value [ value ]* [ # commentString ] \n
blankLine :: \n
commentString :: char [ char ]*

attributeName :: any valid attribute name in the configuration file for the current system.

absoluteAddress :: { decimalAddr | hexAddr }

decimalAddr :: a decimal number in the range 65536 to 134283263

hexAddr :: a hexadecimal format number in the range 0x10000 to 0x0800FFFF

decValue :: a decimal number in the range 0 - 65535

hexValue :: a hexadecimal format number in the range 0x0000 to 0xFFFF

Example: ######################################################
## PROJECT     : ALADDIN
## TITLE       : SEQUENCER MPU LOAD FILE FOR ALADDIN TESTS
## VERSION     : 003
## DATE        : 08/06/2003
## AUTHOR      : KAVI CHOPRA,JOHN GARCIA,PETER MOORE,GUSTAVO RAHMER
## HISTORY     : 07/11/2003 ORIGINAL VERSION.
##             : 07/21/2003 sources aligned to /monsoonhome/aladdin_test.
##             : 07/25/2003 Version for Fowler sampling + Digital averages
##             : 08/06/2003 Allows normal and row reset reads
## DESCRIPTION : LOADS THE SEQUENCER MPU PROGRAM AND PATTERN MEMORY
##             : SEGMENTS FOR THE ALADDIN II ARRAY.
## RESOURCES   :
######################################################
```
# ALADDIN PATTERN MEMORY SEGMENT
# First Give start address
# addr=0x00011000  # you can use absolute address or
# addr=MCB_SEQPATMEM # DHE attribute name or
# addr=mcbSeqPatMem # PAN attribute name
# Now list the values to be loaded into the memory
# you can list one per line or several per line
0x0001 0x0000 0x0002 0x0000 0x0004 0x0000 0x0000 0x0000
0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x2540
53 32456 18