SWIFT ARCVIEW COMMAND REFERENCE

Document Number SWF-MAN-SCI-04
Author ..... Fraser Clarke
Revision: 2.0
Date 11/10/2016
Status ..... Released

CHANGE HISTORY

Version	Date	Author	Change
1.0	16/1/2013	FJC	First release
1.1	2/11/2015	FJC	Small updates to motor control
2.0	11/10/2016	FJC	Moved into word format. Removed obsolete sections

CONTENTS

LIST OF FIGURES

NO TABLE OF FIGURES ENTRIES FOUND.

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AO	Adaptive Optics
CCD	Charge Coupled Device
DCS	Detector Control System
DM	Deformable Mirror
FM3	Fold Mirror ‘3’ (actually the first mirror inside SWIFT)
GUI	Graphical User Interface
HODM	High Order Deformable Mirror
ICS	Instrument Control System
IFU	Integral Field Unit
IFS	Integral Field Spectrograph
LODM	Low Order Deformable Mirror
P3K	See PALM3K
PA	Position Angle (angle of minor axis of field North through East)
PALM3K	Palomar Adaptive Optics System (post-2011)
PALAO	Palomar Adaptive Optics System (pre-2011)
RTD	Real Time Display
SDSU	San Diego State University
Spaxel	SPAtial PIXel, referring to a single spatial pixel in the reconstructed data cube, to differentiate it from a single pixel on the CCD detector
SSM	Star Select Mirror
TBD	To be decided
TCS	Telescope Control System
TO	Telescope Operator
TTM	Tip-Tilt Mirror
VI	Virtual Instrument

1 APPLICABLE AND REFERENCE DOCUMENTS

1.1 Applicable Documents

The following documents at their indicated revision form part of this document to the extent specified herein.

AD1. None

1.2 Reference Documents

The following documents provide useful reference information associated with this document. These documents are to be used for information only. Changes to the date and/or revision number do not make this document out of date

RD1. None

2 ARCVIEW OVERVIEW

ArcVIEW is a set of LabVIEW VIs developed to provide a central command/response server for astronomical instrumentation. It was originally developed for detector control at the SOAR telescope in Chile, and adopted by Palomar as an observatory standard in the late 2000s.

ArcVIEW provides a central server core and a set of utility functions. Modules are built (in LabVIEW) to communicate with this core, enable commands and information to be passed between modules. In SWIFT, ArcVIEW is used to provide central control of instrument system with the ironic exception of the detectors… Modules have been developed for all the hardware elements of the instrument control (e.g. temperature monitors, power supplies, etc etc). This document lists the commands applicable to each of these modules.

In general, most modules are abstracted (via shell scripts) to simple one-line commands which can be issued from the command line (e.g. “arcs on”). Direct communication with ArcVIEW modules is usually only necessary for troubleshooting or engineering tasks.

3 SOFTWARE START-UP

To start ArcVIEW running on swiftlcu, click on the ArcVIEW icon on the desktop, or type;

```
start_ArcVIEW_Server &
```

This will bring up an xterm, which will in turn launch a LabVIEW window running the ArcVIEW programme. ArcVIEW takes ~30 seconds to initialize. The red text in the top left of the window will say “Ready” when it is fully started. ArcVIEW requires X to be running, so if you need to start it over a terminal connection (ssh) on a headless machine, do;

```
export DISPLAY=:0.0
```

The windows will appear on the headless machine’s display.

4 COMMUNICATIONS

Communications from external programmes with ArcVIEW are standard ASCII commands via TCP. For SWIFT, the standard host is `swiftlcu.palomar.caltech.edu` and the standard port is `2040`. On the SWIFT machines, these are defined as the shell variables `$archost` and `$arcport` respectively. `$arctimeout` is also defined, giving the command time-out in milliseconds.

The sendsockcmd programme gives a simple command line interface to ArcVIEW with the following syntax;

```
sendsockcmd –h $archost –p $arcport –t $arctimeout “MODULE {Command} {Parameters}”
```

where `MODULE` is the name of the ArcVIEW module being called. For most modules, simplified wrapper scripts exist in the `/home/swift/bin` directory on both `swiftlcu` and `swiftws`, which provide a quicker way of calling `sendsockcmd` (see Table 1 “summary of modules” below for the shortcuts). The scripts are typically just the name of the module in lower case, e.g. for the P3K module;

```
p3k {COMMAND} {Parameters}
```

In both cases, the response from ArcVIEW has the following general form

```
OSWIFT: MODULE {Answer}
```

Where `MODULE` is again the module name.

5 ERROR RESPONSES

The list below includes common error messages you may see when calling ArcVIEW commands.

5.1 Error 111 writing_message

This usually means ArcVIEW in not running. Sometimes ArcVIEW appears to start OK, communications fail. Solution seems to be to restart ArcVIEW.

5.2 Error MODULE not connected

Either the MODULE in question wasn't loaded when ArcVIEW started up check (check `/usr/local/ArcVIEW/config/APP_AUTOMODULES.cfg`), or the SYNC module which handles the communications hasn't started properly. Run the command;

```
Start_ArcVIEW_Comms
```

which should fix the latter.

5.3 Hangs without timeout

This is a bug within ArcVIEW which causes the TCP server to crash/hang indefinitely. Only known fix is to restart ArcVIEW. Seems to be caused by too frequent calls to ArcVIEW. Under investigation.

6 MODULES

This section provides a description of all the SWIFT specific modules used in ArcVIEW.

Table 1 : Summary of SWIFT ArcVIEW modules

Module	Command line short-cut	Description
BLADE	`blade`	Controls the blade shutter at the exit of the SWIFT slicer. For high contrast work.
BOCADC	`bocadc`	Talks to the pressure monitor to measure cryostat pressures. Currently offline.

ESP300	`esp300`
	`scale`
	`input`
	`mask`
		Talks to the motor controller for the pre-optics scale changer, focal plane mask wheel, and input selection wheel.

FITSheader	`fitsheader`	Generates a FITS header dataset for the DCS.
IPPOWER	`ippower` `arcs`	Talks to the network power supply and controls eight switches, including the arc lamps (there is a specific short-cut to turn on/off arcs)
LS218	`ls218`	Communicates with the LakeShore218 temperature monitor
LS325	`ls325`	Communicates with the LakeShore325 temperature controller, which monitors/controls the CCD temperatures
P200TCS	`tcs`	Communicates with the Palomar 200-inch telescope control system
P3K	`p3k`	Communicates with the P3K AO system
TTIPSU	`ttipsu` `halogen`	Controls the power supply which powers the halogen lamp
WEATHER	-	Monitors the Palomar weather status

6.1 Commands and Parameters

The module descriptions below outline the commands and associated parameters for each module. Commands are given in CAPITALS and parameters (e.g. the user's choice) are given in Title format. Optional parameters are enclosed in {braces}.

6.2 BLADE

6.2.1 Description

This module controls the servo motor which is attached to the blade shutter at the exit of the image slicer. Adjusting the position of the blade allows the user to block 0-8 central slices on the IFU, to prevent saturation when observing bright stars.

The module communicates with a virtual USB port provided by usbclnt. The physical port is on the dumb machine swiftic

6.2.2 Configuration

6.2.2.1 Module configuration

The module configuration file is held in `/usr/local/ArcVIEW/config/MOD_BLADE.cfg`

This defines the communication settings for the module, i.e. the port for the usbclnt connection to swiftic.

6.2.2.2 Motor configuration

The motor configuration file is `/usr/local/ArcVIEW/config/BLADE_MotorConfig.cfg`

This defines the positions (in angle) of all of the blade options

6.2.3 Commands

`CONNECT`

Make a connection to the usbclnt port

`DISCONNECT`

Drop the connection to the usbclnt port

`GET`

Returns the current position of the blade in degrees

`SET Position`

Sets the position of the blade to a given position (as defined in the motor configuration file) or angle. Position options are;

- **Open** – blade completely out of the beam
- **OneSlit** – one slit covered

- **TwoSlit** – two slits covered
- …
- **EightSlit** – eight slits covered

6.3 BOCADC

Currently offline due to an unidentified memory leak bug

6.3.1 Description

Reads the pressures of the SWIFT cryostats from the BOC ADC pressure monitor

6.3.2 Configuration

6.3.3 Commands

6.4 ESP300

6.4.1 Description

This module provides an interface to the Newport ESP300 motor controller. This controls the three main motions in the instrument; the pre-optics scale, the focal plane mask, and the input selection stage. There are command line short cuts for each of the three main motions;

- `scale 235mas/80mas/16mas` - Set the spaxel scale
- `input Telescope/Calibration` - Set the input fold mirror position
- `mask Open/PinHole/CoarseVerticalLines/FineVerticalLines/etc` - Set the focal plane mask position

6.4.2 Configuration files

6.4.2.1 Module configuration

The module configuration is held in `/usr/local/ArcVIEW/config/MOD_ESP300.cfg`

This file defines the serial port which communicates with the motor controller, and the motor configuration file.

6.4.2.2 Motor configuration

The motor configuration is held `/usr/local/ArcVIEW/config/ESP300_MotorConfig.cfg`

This file defines the names and angles of all the predefined positions. The name of this file can be defined via the MotorConfigFile option in the module configuration file.

6.4.3 Commands

```
MOVE AXIS 1/2/3 Position/Angle
```

Moves the specified axis to the specified postion (string matching a predefined position in the config file) or angle (arbitrary number). If an unknown position is requested, the module will return an error.

```
GET AXIS 1/2/3 {POSITIONS}
```

Returns current status of the specified axis. The format is; `Angle (PositionName) Power`.

If “POSITIONS” is added to the call, the module returns the available predefined position names for that axis.

```
POWER ON/OFF {AXIS 1/2/3}
```

Turn on/off the power for all (default) or one (specified) axis.

```
INIT
```

Initalise all motor speeds to maximum values (except axis 1, which is set to 10 deg/s).

```
HOME AXIS/ALL {1/2/3}
```

Home one or all of the axes. If AXIS is passed as an option, the axis number must be specified.

```
PASS {Command}
```

Passes a command directly to the ESP300 controller. Engineering use only.

```
RATE {?/Value} (TBC)
```

Sets or returns the update rate of the status checking. If “?” or no option is passed, it returns the rate. If a numeric value is passed (in milliseconds), the update rate is changed, within predefined limits.

```
STATE
```

Returns the current state of the module

```
SHUTDOWN
```

Shuts down the module.

6.5 FITSheader

6.5.1 Description

This module generates a FITS header once per second, based on the current status of the instrument database. The FITS header is written out as text file, which is read and included in the image by the DCS software.

6.5.2 Configuration

6.5.2.1 Module configuration

The module configuration is held in `/usr/local/ArcVIEW/config/MOD_FITSheader.cfg`

This file defines the header template file and the location of the header text file.

6.5.2.2 Header configuration file

The header template file is (by default) held `/usr/local/ArcVIEW/config/SWIFTheader.tpl`

This file defines the structure of the header by listing the header keywords and the corresponding instrument database parameter. The name of this file can be defined via the template option in the module configuration file.

6.5.3 Commands

`STATE`

Returns the current status of the module

`GET`

Returns the current fits header as a string, with a maximum length of 1028 characters. If the header exceeds this length (usually it does), subsequent calls to “GET” will return the remaining characters.

`SHUTDOWN`

Stops the module running.

6.6 IPPOWER

6.6.1 Description

This module communicates with the IP9258 internet power supply module. It is used to turn on/off the arc lamps, and can also be used to power on/off the SDSU controllers and remote computers. Note that there is a specific command line short-cut to call this module and turn on/off the arcs. It is;

```
arcs on/off
```

6.6.2 Configuration

6.6.2.1 Module configuration

The module configuration file is held in `/usr/local/ArcVIEW/config/MOD_IPPOWER.cfg`

This defines the communication settings for the module, i.e. the hostname and user for swiftnps, the network power server (NB, the password for swiftnps is stored within the module code).

6.6.2.2 Power switch configuration

The power switch configuration file is `/usr/local/ArcVIEW/config/IPPOWER_PowerConfig.cfg`

This defines the names, default states and alarms of each of the eight switches in the network power supply.

6.6.3 Commands

`STATE`

Returns the current state of the module

`SHUTDOWN`

Stop the module running.

`POWERON Socket#`

Turns the power off to a socket. Accepts either the name of the socket, or the number of the socket from 1 to 8.

`POWEROFF Socket#`

Turns the power on to a socket. Accepts either the name of the socket, or the number of the socket from 1 to 8.

`GETSOCKETS`

Returns a list of the socket names as a comma separated list.

`GETPOWER {All/#}`

Returns the power status all sockets, or of a specific socket numbered from 1 to 8. Returned value is ON or OFF.

6.7 LS218

6.7.1 Description

Reports temperatures from the LakeShore218 temperature monitor

6.7.2 Configuration files

6.7.2.1 Module configuration

The module configuration file is held in `/usr/local/ArcVIEW/config/MOD_LS218.cfg`

This defines the communication settings for the module, i.e. the port to connect to swiftts on, and the sensor configuration file

6.7.2.2 Sensor configuration

The sensor configuration file is held in `/usr/local/ArcVIEW/config/LS218_GaugeNames.cfg`

This file defines the names of each of the eight temperature sensors (“gauges”) read by the LS218 hardware. It also defines alarms on certain sensors, though these are not currently fully implemented in the software.

6.7.3 Commands

```
STATE
```

Returns the current state of the module.

```
SHUTDOWN
```

Stops the module running.

```
GET
```

Returns the current temperatures as read from the LakeShore monitor as a comma separated string.

```
PASS {Command}
```

Sends a command directly to the LS218 hardware. Engineering use only.

```
RATE {?/Value}
```

Sets of returns the update rate of the module. Passing “?” or no argument returns the current rate. Passing a numeric value (in milliseconds) sets the rate, within predefined limits

6.8 LS325

6.8.1 Description

Provides an interface to the LakeShore325 temperature monitor and controller, which monitors and controls the temperature of the SWIFT CCDs

6.8.2 Configuration files

6.8.2.1 Module configuration

The module configuration file is held in `/usr/local/ArcVIEW/config/MOD_LS325.cfg`

This defines the communication settings for the module, i.e. the port to connect to swiftts on, and the sensor configuration file

6.8.2.2 Sensor configuration

The sensor configuration file is held in `/usr/local/ArcVIEW/config/LS325_GaugeNames.cfg`

This file defines the names of two temperature sensors (“gauges”) read by the LS325 hardware.

6.8.3 Commands

`STATE`

Returns the current state of the module

`SHUTDOWN`

Stops the module running

`GET`

Returns the current temperatures as read from the controller as a comma separated string.

`PASS {Command}`

Sends a command directly to the LS218 hardware. Engineering use only.

`RATE {?/Value}`

Sets of returns the update rate of the module. Passing "?" or no argument returns the current rate. Passing a numeric value (in milliseconds) sets the rate, within predefined limits

6.9 P200TCS

6.9.1 Description

This module provides an interface to the Palomar 200-inch TCS. Note that the short-cut for sending commands is just "`tcs`".

6.9.2 Configuration

Module configuration is stored in the file `/usr/local/ArcVIEW/config/MOD_P200TCS.cfg`

This file defines the TCS communication parameters, and the offset angle between the Cassegrain ring angle and the SWIFT position angle (-260 degrees since 2011).

6.9.3 Commands

`STATE`

Returns the current state of the module. No options.

`SHUTDOWN`

Shuts down the module. No options.

`RATE {?/Value}`

Sets or returns the current update rate of the TCS status requests. Passing "?" or no argument returns the current rate. Passing a numeric value (milliseconds) changes the update rate (within predefined limits).

`CONNECT`

Enables TCS connection. No options

```
DISCONNECT
```

Disables TCS connection. No options

```
OFFSET ra dec
```

Send an offset command to the TCS (via ‘pt’). The offsets are provided in arcseconds with East and North positive.

```
TRACKINGRATES ra dec
```

Sets the tracking rates of the telescope to the given values in RA and DEC (via TCS ‘rates’ command).

```
NONSIDERIAL ON/OFF
```

Turns on or off nonsiderial tracking (via TCS ‘r’ command).

```
RING Angle
```

Moves the cassegrain rotator to a given angle (via TCS ‘ringo’ command). Note, this sets the cassegrain ring angle NOT the swift position angle, so you must account for the offset (currently +100 degrees), or use the PA command.

```
PA Angle
```

Not yet tested!

Moves the cassegrain rotator to place the SWIFT field at given PA (PA=0 gives North along the short axis of the IFU field, with the master chip on the North).

```
COORDS Ra Dec Equinox RaMotion DecMotion MotionFlag ObjectName
```

Passes a coordinate string to the TCS in the standard P200 format. Requires Ra in decimal hours and Dec in decimal degrees. Rates must be passed for RaMotion and DecMotion, even if zero. MotionFlag specifies the type of motion; 0=proper motions, 1=tracking rates (arcsec/hr), 2=tracking rates (sec/hr & arcsec/hr).

Object name must also be supplied as a string.

```
RET
```

Passed directly to TCS. No options

```
Z
```

Passed directly to TCS. No options

```
GO
```

Pops up a query box asking the user to confirm the telescope has been moved to the new coordinates. Intended for future script operation.

```
TX
```

Not implemented (observer shouldn’t do this)

6.10 P3K

6.10.1 Description

This module provides the instrument interface to the P3K AO system

6.10.2 Configuration

Module configuration is stored in the file `/usr/local/ArcVIEW/config/MOD_P3K.cfg`

This file defines the communication parameters to the AO system (host addresses and ports).

6.10.3 Commands

`STATE`

Returns the current state of the module. No options.

`SHUTDOWN`

Shuts down the module. No options.

`RATE {?/Value}:`

Sets or returns the current update rate of the P3K status requests. Passing "?" or no argument returns the current rate. Passing a numeric value (milliseconds) changes the update rate (within predefined limits).

`DM {ON/OFF/STATUS}:`

Not yet implemented fully

Sets or returns the current status of the deformable mirrors. Passing "?", "STATUS" or no argument returns the current status of the deformable mirrors, as of the last regular status update from the AO system. Sending "ON" or "OFF" turns the deformable mirror on or off respectively (TBC – might need some error checking in here to make sure the TTM is on first)

`TTM {ON/OFF/STATUS/?}`

Not yet implemented fully

Sets or returns the current status of the tip-tilt mirror. Passing "?", "STATUS" or no argument returns the current status of the tip-tilt mirror, as of the last regular status update from the AO system. Sending "ON" or "OFF" turns the tip-tilt mirror on or off respectively (TBC – might need some error checking in here to make sure the DM is off first)

`TCS {ON/OFF/STATUS/?}`

Not yet implemented

Sets or returns the current status of the AO's link to the TCS.

`STIM_SELECT {SKY/WLIGHT/?}`

Sets or returns the current status of the AO Stimulus source. Passing "?" or no argument returns the current status. Passing WLIGHT or SKY moves the stimulus either in or out of the beam

respectively. This takes time, so the user should check that sci_motion is complete before continuing.

```
SCI_INST {SWIFT}
```

Sets the AO science instrument to SWIFT. This loads the relevant parameters from the AO configuration file (e.g. instrument PA, default focus, etc). This should be issued automatically on start-up. There doesn't seem to be a way to check which is the current instrument though, so it is worth check for an "ACK" response to this command…

```
SCI_MOTION {N/S/E/W/U/D/L/R arcseconds} {radec ra_arcseconds dec_arcseconds} {no_ssm}
```

Offset the object on the science field of view & keep it locked in the HWFS. Just passes to AO. Maybe should rewrite this to allow easier off-set skies by opening loops first.

6.11 TTIPSU

6.11.1 Description

This module interfaces with the Thurby-Thander power supply mounted in the SWIFT rack. Its main job is to control the voltages for the Halogen flat-field lamp. You can use either "`ttipsu`" or "`halogen`" on the command line; they are the same thing.

6.11.2 Configuration

Module configuration is stored in the file `/usr/local/ArcVIEW/config/MOD_TTIPSU.cfg`

This file defines the communication parameters (i.e. the port on swifts), the power supply channel in use, and the maximum allowed voltage.

6.11.3 Commands

```
GET
```

Returns the current voltage and current of the lamp

```
ON/OFF
```

Turns the lamp on or off

```
SET Preset
```

Sets the voltage to preset value based on the plate scale. Options are;

- 235mas
- 80mas
- 16mas

Note that this does not turn the voltages on! You must issue "`ttipsu on`" (or "`halogen on`") separately.

```
SHUTDOWN
```

Stops the module running

6.12 WEATHER

6.12.1 Description

This module reads the weather status from the Palomar weather webpage, and writes it into the SWIFT instrument database.

6.12.2 Configuration

The module configuration file is held in `/usr/local/ArcVIEW/config/MOD_WEATHER.cfg`

This defines the URL of the Palomar weather webpage, which is used as the source of the weather information for the module.

6.12.3 Commands

`SHUTDOWN`

Stops the module running.