PlaneWave Interface 4 API

Introduction

PlaneWave Interface 4 (PWI4) is a software package that can control a variety of PlaneWave equipment, including L-series direct drive mounts and the PW1000 1-meter telescope system. These hardware devices natively provide a fairly basic set of capabilities, such as reading the current position of a motor and commanding a desired velocity. PWI4 extends these capabilities by performing astronomical coordinate transformations, measuring and applying pointing model corrections, calculating field rotation effects, following trajectories for a variety of targets including satellites, and much more.

User applications will generally communicate with PlaneWave equipment using PWI4 as an intermediary.

Interfaces

It is currently possible for applications to communicate with PWI4 in two ways:

1. ASCOM drivers (Windows-only)
2. PWI4-hosted HTTP server

ASCOM drivers

ASCOM provides a standard driver model for allowing Windows-based astronomy applications to communicate with astronomical devices, such as mounts, focusers, rotators, and cameras. A wide variety of applications can use ASCOM-compatible devices, including TheSkyX, MaxIm DL, FocusMax, ACP, CCDAutoPilot, Sequence Generator Pro, and more. The PWI4 ASCOM drivers allow all of these applications, along with any ASCOM-compatible applications developed in the future, to use PlaneWave equipment out of the box.

The ASCOM drivers are in fact fairly simple wrappers around the HTTP interface discussed below. Because ASCOM only supports capabilities that are fairly common across all astronomical equipment, the PWI4 ASCOM drivers expose only a subset of the full capabilities provided by PWI4. For custom-developed applications, it may make more sense to use the HTTP interface directly.

For more details on working with ASCOM drivers, refer to https://ascom-standards.org/. Please contact PlaneWave Instruments for sample code that illustrates how to talk to the PWI4 ASCOM drivers.

HTTP server

When PWI4 is running, it hosts an HTTP (web) server that can be used to issue commands to PWI4 and read the status of the system. Essentially every programming language in common use is able to make HTTP requests, either using a built-in construct (e.g. urllib in Python, HttpURLConnection in Java, WebRequest in .NET) or a third-party library/tool (e.g. libcurl for C, curl for shell scripts). Users can

experiment with commands simply by entering URLs in a web browser, and could potentially create a custom user interface to their telescope using HTML and Javascript.

HTTP Server request/response format

By default, the HTTP server is hosted on port 8220.

A request to PWI4 will consist of a URL with the following format:

http://host:port/subsystem/command?param1=value1&param2=value2&...

host: The hostname or IP address of the machine running PWI4. This will often be **localhost** for the common case where the client program is running on the same machine as PWI4.

port: The TCP port number that the PWI4 HTTP server is listening on. By default this is **8220**.

/subsystem/command: The command that is being sent to PWI4. In many cases both a subsystem and a command are specified (for example, **/mount/stop** and **/focuser/goto**), but in some cases no subsystem is specified (for example, **/status**).

param1=value1: For commands that take parameters, one or more parameters can be specified as name=value pairs. For example, the **/mount/goto_ra_dec_j2000** command takes parameters named **ra_hours** and **dec_degs**, and each parameter takes a numeric value.

As a basic example, the following URL retrieves the status from a copy of PWI4 running on the local machine:

http://localhost:8220/status

Here is an example of a request with a two-part command (specifying both the subsystem and the action) that takes parameters:

http://localhost:8220/mount/goto_ra_dec_j2000?ra_hours=10.5&dec_degs=78.321

For most commands, the response consists of a series of keyword=value pairs separated by newlines (ASCII character 0x0A, commonly represented as “\n”). A sample response is shown below:

```
pwi4.version=4.0.9 beta 15
pwi4.version_field[0]=4
pwi4.version_field[1]=0
pwi4.version_field[2]=9
pwi4.version_field[3]=15
response.timestamp_utc=2021-03-11 17:59:43.925011
site.latitude_degs=33.4999722222222
site.longitude_degs=-118
site.height_meters=50
site.lmst_hours=21.4366499466139
mount.is_connected=true
mount.geometry=0
mount.timestamp_utc=2021-03-11 17:59:43.8398
mount.julian_date=2459285.24981295
mount.slew_time_constant=0.1
mount.ra_apparent_hours=10.533857800413
mount.dec_apparent_degs=78.2144328780701
mount.ra_j2000_hours=10.4999990351138
mount.dec_j2000_degs=78.3210015292187
mount.target_ra_apparent_hours=10.5338580142121
mount.target_dec_apparent_degs=78.2144331010628
mount.azimuth_degs=356.418076719304
mount.altitude_degs=22.1858968987629
mount.is_slewing=true
mount.is_tracking=true
mount.field_angle_here_degs=14.7538253824039
mount.field_angle_at_target_degs=14.7538282912822
mount.field_angle_rate_at_target_degs_per_sec=-0.00374925895093006
mount.path_angle_at_target_degs=104.753623319249
mount.path_angle_rate_at_target_degs_per_sec=-0.000371057929011126
mount.axis0.is_enabled=true
mount.axis0.rms_error_arcsec=0.000800749584963323
mount.axis0.dist_to_target_arcsec=2.05761023451979E-05
mount.axis0.servo_error_arcsec=0
mount.axis0.position_degs=356.41814717262
mount.axis0.position_timestamp=2021-03-11 17:59:43.9191
mount.axis1.is_enabled=true
mount.axis1.rms_error_arcsec=0.000381576721903539
mount.axis1.dist_to_target_arcsec=-0.000273245186655727
mount.axis1.servo_error_arcsec=0
mount.axis1.position_degs=22.1858798084183
mount.axis1.position_timestamp=2021-03-11 17:59:43.9191
mount.model.filename=
mount.model.num_points_total=0
mount.model.num_points_enabled=0
mount.model.rms_error_arcsec=0
focuser.is_connected=false
focuser.is_enabled=false
focuser.position=0
focuser.is_moving=false
rotator.is_connected=false
rotator.is_enabled=false
rotator.mech_position_degs=0
rotator.field_angle_degs=0
rotator.is_moving=false
rotator.is_slewing=false
m3.port=0
```

```
autofocus.is_running=false
autofocus.success=false
autofocus.best_position=0
autofocus.tolerance=0
```

Status format

Keywords in the response are generally divided into [category].[property] pairs. Values are generally one of the following types:

Floating point: Can be either standard decimal notation, such as "-1.234", or scientific notation to express very large or very small values, such as "6.294E-08".

Boolean: Can be either "true" or "false"

Integer: A simple series of digits with an optional sign, such as "180" or "-1".

String: An arbitrary sequence of printable characters, such as "PlaneWave CDK700" or "-12:34:56.7".

Timestamp: A string of the form:

yyyy-MM-dd HH:mm:ss.ffffff

yyyy: The 4-digit year (e.g. 2021)
MM: The 2-digit month (00 – 12)
dd: The 2-digit day (00 – 31)
HH: The 2-digit hour (00 – 23)
mm: The 2-digit minute (00 – 59)
ss: The 2-digit second (00 – 59)
ffffff: The fractional part of a second (000000 – 999999)

Status values

The values included in a status response are described below.

Name	Datatype	Description
pwi4.version	string	The version of PWI 4 that is currently running. Version numbers are structured as follows: [major].[minor].[revision] [beta_num_or_final]

		where: [major] will always be 4 for the PWI 4 series of control software [minor] is incremented when substantial new features or hardware support are added [revision] is incremented when smaller additions or bugfixes are made [beta_num] may **optionally** contain the word "beta " followed by a number; for example: "beta 13". Versions with features that may not have been extensively tested yet typically have the "beta" tag. After awhile, these versions get promoted to "final" releases, which do not contain the "beta" tag. Sample values: "4.0.8" "4.0.9 beta 13"
pwi4.version_field[0]	int	The "major" portion of the PWI4 version. For PWI 4, this should always be "4".
pwi4.version_field[1]	int	The "minor" portion of the PWI4 version.
pwi4.version_field[2]	int	The "revision" portion of the PWI4 version
pwi4.version_field[3]	int	The "beta" iteration of the PWI4 version. If this release is a "final" (non-beta) version, this field will have the value 99.
response.timestamp_utc	timestamp	The UTC time, according to the PC clock, at which the text of this status response was generated. This value can be compared to the time at which the client initiated the request or received the response (if those times are derived from the same PC clock, or another clock that is synchronized to the same source as the PC clock) to measure the overhead in transmitting the request or response. This value can also be compared to other timestamps included in a status response to determine the age of various telemetry values reported by the system.
site.latitude_degs	float	The configured latitude of the telescope, in degrees. Negative values are south of the equator.

		The expected range is from -90 to +90 degrees.
site.longitude_degs	float	The configured longitude of the telescope, in degrees. Negative values are west of the prime meridian. The typical range will be from -180 to +180 degrees, although values from -360 to +360 are allowed.
site.height_meters	float	The configured height (elevation) of the telescope above sea level, in meters.
site.lmst_hours	float	The local mean sidereal time at the telescope, in decimal hours. The expected range is from 0 to 24 hours.
mount.is_connected	boolean	“true” if PWI4 has an active connection to the mount hardware.
mount.geometry	int	The geometry of configured mount. 0: Alt-Az 1: Equatorial Fork 2: German Equatorial
mount.timestamp_utc	timestamp	The UTC time (according to the PC clock) at which the status values of the mount were last updated. If PWI4 is not connected to the mount, the value will be 0001-01-01 00:00:00.0000
mount.ra_apparent_hours	float	The telescope’s current position in hours of apparent right ascension. Pointing model corrections (if present) have been applied. Relative to a J2000 target, apparent coordinates account for precession, nutation, and annual aberration, but not refraction. If PWI4 is not connected to the mount, the value will be 0.
mount.dec_apparent_degs	float	The telescope’s current position in degrees of apparent declination. Pointing model corrections (if present) have been applied. Relative to a J2000 target, apparent coordinates account for precession, nutation, and annual aberration, but not refraction. If PWI4 is not connected to the mount, the value will be 0.
mount.ra_j2000_hours	float	The telescope’s current position in hours of J2000 right ascension. Pointing model corrections (if present) have been applied. If PWI4 is not connected to the mount, the value will be 0.
mount.dec_j2000_degs	float	The telescope’s current position in degrees of J2000 declination. Pointing model corrections (if

		present) have been applied. If PWI4 is not connected to the mount, the value will be 0.
mount.target_ra_apparent_hours	float	The position of the target the telescope is trying to acquire or follow, in hours of apparent right ascension. As soon as a new target is submitted, this value (and mount.target_dec_apparent_degs) will update to reflect the new target coordinates. This value includes the baseline target coordinates as well as the total accumulated RA/Dec and Path/Transverse offsets. As of PWI 4.0.9, it does not incorporate the native Axis0/Axis1 offsets, but this may be subject to change in the future. If PWI4 is not connected to the mount, or if the mount is not currently following a target, this value will be 0.
mount.target_dec_apparent_degs	float	The position of the target the telescope is trying to acquire or follow, in degrees of apparent declination. Refer to mount.target_ra_apparent_hours for more information. If PWI4 is not connected to the mount, or if the mount is not currently following a target, this value will be 0.
mount.azimuth_degs	float	The telescope's current position in degrees of azimuth. North is defined as 0 degrees, and East is 90 degrees. This value incorporates pointing model corrections. For the raw reading of the Azimuth motor position in an Alt/Az mount, refer to mount.axis0.position_degs. If PWI4 is not connected to the mount, this value will be 0.
mount.altitude_degs	float	The telescope's current position in degrees of altitude. 0 degrees is the horizon, and 90 degrees is zenith. This value incorporates pointing model corrections. For the raw reading of the Altitude motor position in an Alt/Az mount, refer to mount.axis1.position_degs. If PWI4 is not connected to the mount, this value will be 0.
mount.is_slewing	Boolean	When the mount is first commanded to follow a

		new target, this flag reports true while the mount is moving to acquire the target. Once the target is acquired, or if the movement is stopped, this flag reports false. If PWI4 is not connected to the mount, the value will be false.
mount.is_tracking	boolean	When the mount is trying to follow a target, this flag reports true. If mount.is_slewing is also true, then the mount has not yet acquired the target. When the mount is stopped, this flag reports false.
mount.field_angle_here_degs	float	The amount of field rotation induced by the geometry of the mount at the current telescope position. For equatorial fork mounts with good polar alignment, this number will be close to 0. For Alt-Az mounts, this number will change depending on where the telescope is pointing. On a well-leveled Alt-Az mount, this number will correlate closely with the parallactic angle at the current pointing position. Because this number reports the calculated field rotation angle for the last-sampled position of the telescope, this number may change dramatically over the course of a slew to a new target. In many cases, the value mount.field_angle_at_target_degs is more useful since it reports what the field rotation angle will be once the telescope has arrived at its target. This value incorporates corrections from the pointing model if one is present. This value only accounts for amount of field rotation caused by the mount. The effects of an instrument rotator are handled separately.
mount.field_angle_at_target_degs	float	The amount of field rotation induced by the geometry of the mount at the target (destination) position of the telescope. For equatorial fork mounts with good polar alignment, this number will be close to 0. For Alt-Az mounts, this number will change

		depending on where the target is located in the sky. On a well-leveled Alt-Az mount, this number will closely correlate with the parallactic angle at the target position. On an Alt-Az mount, this value will typically jump to a new value when a new target is requested, and then will slowly change as the target moves across the sky. This value incorporates corrections from the pointing model if one is present. This value only accounts for amount of field rotation caused by the mount. The effects of an instrument rotator are handled separately.
mount.field_angle_rate_at_target _degs_per_sec	float	The rate at which mount.field_angle_at_target_degs is changing, in degrees per second.
mount.path_angle_at_target_degs	float	
mount.path_angle_rate_at_target _degs_per_sec	float	
mount.axisN.*		Contains status information about the motorized axes controlled by the mount axis0: The primary axis of rotation. Corresponds to the Azimuth axis in an Alt-Az mount, or the Right Ascension axis in an Equatorial mount. axis1: The secondary axis of rotation. Corresponds to the Altitude axis in an Alt-Az mount, or the Declination axis in an Equatorial mount.
mount.axisN.is_enabled	boolean	**true** if the motor for this axis is powered on and under servo control **false** if the motor is not under servo control For a direct-drive motor, an enabled motor will resist being pushed out of position, and a disabled motor can normally be moved freely by hand. An enabled motor may transition to the disabled state if an error is detected in the control system – for example, if the motor is using too much current, or is too far away from its expected position.
mount.axisN.rms_error_arcsec	float	The root-mean-square value of the recent mount.axisN.dist_to_target_arcsec

		measurements. Provides a basic estimate of how well the motor is tracking its intended target. If the PWI4 is not connect to the mount, the motor is disabled, or the motor is not following a target, this value will be 0.
mount.axisN.dist_to_target_arcsec	float	The distance from the motor's last-sampled position and the desired position of the axis at that time. When a new distant target is submitted, this value will be large at first and will gradually towards zero as the motor approaches the target.
mount.axisN.servo_error_arcsec	float	The distance from the desired position of the motor and the sampled position of the encoder, as reported by the servo control electronics.
mount.axisN.position_degs	float	The raw position of the motor encoder, in degrees
mount.axisN.position_timestamp	float	The UTC time when the axis telemetry was sampled
mount.model.filename	string	The name of the pointing model file that is currently loaded. If no model is loaded, this value will have zero length.
mount.model.num_points.total	int	The total number of calibration points in the active pointing model.
mount.model.num_points.enabled	int	The number of calibration points in the active pointing model that are contributing to the model fit.
mount.model.rms_error_arcsec	float	The root-mean-square error of the calibration points compared to the current best fit of the model
focuser.is_connected	boolean	
focuser.is_enabled	boolean	
focuser.position	float	
focuser.is_moving	boolean	
rotator.is_connected	boolean	
rotator.is_enabled	boolean	
rotator.mech_position_degs	float	
rotator.field_angle_degs	float	
rotator.is_moving	boolean	
rotator.is_slewing	boolean	
m3.port	int	
autofocus.is_running	boolean	
autofocus.success	boolean	
autofocus.best_position	float	
autofocus.tolerance	float	

Commands

/status

/mount/connect

/mount/disconnect

/mount/enable

/mount/disable

/mount/stop

/mount/goto_ra_dec_apparent

/mount/goto_ra_dec_j2000

/mount/goto_alt_az

/mount/goto_coord_pair

/mount/park

/mount/set_park_here

/mount/offset

/mount/tracking_on

/mount/tracking_off

/mount/follow_tle

/mount/find_home

/mount/set_slew_time_constant

/mount/model/add_point

/mount/model/clear_points

/mount/model/save_as_default

/mount/model/save

/mount/model/load

/mount/radecpath/new

/mount/radecpath/show

/mount/radecpath/add_point

/mount/radecpath/apply

/mount/custom_path/new

/mount/custom_path/add_point_list

/mount/custom_path/apply

/focuser/enable

/focuser/disable

/focuser/goto

/focuser/stop

/rotator/enable

/rotator/disable

/rotator/goto_mech

/rotator/goto_field

/rotator/offset

/rotator/stop

Python reference implementation

A reference implementation for communicating with PWI4 via the HTTP interface can be found here:

http://planewave.com/files/software/PWI4/python/pwi4_client.py

The code should be compatible with Python 2.7 and Python 3.x.

An example of how to use this Python module can be found here:

http://planewave.com/files/software/PWI4/python/pwi4_client_demo.py

A full list of supported commands can be found by looking through the pwi4_client.py code. For example, if we look at the implementation of the mount_connect() function:

```
def mount_connect(self):
    return self.request_with_status("/mount/connect")
```

we can derive the equivalent HTTP request:

http://localhost:8220/mount/connect

Similarly, if we look at a function that takes some arguments:

```
def mount_goto_alt_az(self, alt_degs, az_degs):
    return self.request_with_status(
               "/mount/goto_alt_az",
               alt_degs=alt_degs,
               az_degs=az_degs)
```

this can be converted into a URL such as the following:

http://localhost:8220/mount/goto_alt_az?alt_degs=45.123&az_degs=315.987

All status values are parsed in the PWI4Status class, and information about the available names and datatypes can be found by looking at that code. For example, by looking at the following section of code:

```
self.mount.axis0 = Section()
    self.mount.axis0.is_enabled =
        self.get_bool("mount.axis0.is_enabled")
    self.mount.axis0.rms_error_arcsec =
        self.get_float("mount.axis0.rms_error_arcsec")
    self.mount.axis0.dist_to_target_arcsec =
        self.get_float("mount.axis0.dist_to_target_arcsec")
    self.mount.axis0.servo_error_arcsec =
        self.get_float("mount.axis0.servo_error_arcsec")
    self.mount.axis0.position_degs =
        self.get_float("mount.axis0.position_degs")
    self.mount.axis0.position_timestamp_str =
        self.get_string("mount.axis0.position_timestamp")
```

we can see that the following status values are available for reading:

Name	Datatype

mount.axis0.is_enabled	boolean
mount.axis0.rms_error_arcsec	floating-point
mount.axis0.dist_to_target_arcsec	floating-point
mount.axis0.servo_error_arcsec	floating-point
mount.axis0.position_degs	floating-point
mount.axis0.position_timestamp	string

A full description of all available commands and status values will be available in an upcoming version of this documentation. In the meantime, please contact PlaneWave Instruments with any questions about the specific definition, interpretation, or recommended use of any commands or values.