

ASRAS Callisto

SunTracker Driver

Protocol Specification
Release 1.0

Author: Brandon McDonnell <bxm3364@rit.edu>

Table of Contents

Revision History.....	2
Introduction.....	3
Purpose.....	3
Server Operation.....	3
Document Conventions.....	3
Client Commands.....	4
status.....	4
cal (tentative/legacy).....	6
calMove (tentative/legacy).....	8
capture.....	9
connect.....	10
environment.....	11
environmentConfig.....	12
fitList.....	14
locationConfig.....	15
mode.....	16
motorConfig.....	17
parkConfig.....	18
position.....	19
positionConfig (tentative).....	21
scheduleCreate.....	22
scheduleDelete.....	23
scheduleList.....	24
target.....	25
Server Updates.....	26
position.....	26

Revision History

Author	Date	Comment
Brandon McDonnell	5 Feb 2018	Draft Release 1.0
Brandon McDonnell	22 Feb 2018	Release 1.0

Introduction

Purpose

The SunTracker Driver is the central control software that runs on the instrument PC and interacts with hardware components, client software, and e-Callisto software. The driver hosts a low-latency WebSocket service that provides a standard interface that any client software on any platform may use to communicate. For example, the LABVIEW client and the ASCOM driver may both communicate over this protocol. This document defines the format and capabilities of the service.

Server Operation

The SunTracker Driver server listens on unencrypted port 8119 by default. Connection to the server is done by WebSocket over HTTP. Communication is bidirectional; clients may send commands to the server and should expect to receive three types of messages: reports, updates, and errors. Messages are encoded with the JSON standard.

Clients may send commands with a unique ID; the server will return this ID in the corresponding response.

Each message defined in this document is accompanied by an example message; if unfamiliar with JSON, it is recommended to review some example messages now.

Any logfiles or databases are stored in the same folder as the Driver executable. Settings may be set or changed by editing the Server's configuration file and restarting the Server, or by sending reconfiguration commands to the server. For more information, see the SunTracker Driver User Manual.

Document Conventions

Italics : This key is optional and may be omitted in the Request, Response, or Notification.

Client Commands

Client commands are sent from the client to the server. The server will provide a response to each command and may produce updates as a result of any action the system takes as a result.

status

Reports the overall system status. Useful for obtaining a high-level status update for the entire system; see other methods for more detailed status reporting.

Request parameters

- none -

Response parameters

Key	Possible Values
driver	“active” – Driver is functioning properly.
	“warning” – Driver has triggered warnings that should be investigated, but driver is still functioning.
	“failure” – Driver has critically failed and is not functional.
callisto1	“not configured” – Callisto is not configured.
	“active” – Callisto is configured and operational.
	“warning” – Callisto is configured and operational but may not be functioning properly.
	“failure” – Callisto is configured but failed.
	“offline” – Callisto is configured but not detected.
callisto2	“not configured” – Callisto is not configured.
	“active” – Callisto is configured and operational.
	“warning” – Callisto is configured and operational but may not be functioning properly.
	“failure” – Callisto is configured but failed.
	“offline” – Callisto is configured but not detected.
hwControl	“not configured” – Hardware controller is not configured; demo mode.
	“active” – Hardware controller is configured and operational.
	“warning” – Hardware controller is configured and operational but may not be functioning properly.

Key	Possible Values
	“failure” – Hardware controller is configured but failed.
	“offline” – Hardware controller is configured but not detected.
lastAlert	[not present] – No failures logged since system startup.
	[number timestamp] – Last failure timestamp.
fileXfer	“not configured” – File upload not configured.
	“active” – File upload is configured and operational.
	“failure” – File upload is configured but not functioning properly.
emailAlert	“not configured” – Email alerts not configured.
	“active” – Email alerts configured and operational.
	“failure” – Email alerts configured but not functioning properly.
stackLight	“not configured” – Stack light is not configured; demo mode.
	“active” – Stack light is configured and operational.
	“warning” – Stack light is configured and operational but may not be functioning properly.
	“failure” – Stack light is configured but failed.
	“offline” – Stack light is configured but not detected.

Errors

No errors can occur during this request.

Example

Request:

```
{ "method": "status" }
```

Response:

```
{ "method": "status", "params": { "driver": "active", "callisto1": "active", "callisto2": "not configured",
"hwControl": "warning", "lastAlert": 1517869870, "fileXfer": "active", "emailAlert": "active",
"stackLight": "active" } }
```

cal (tentative/legacy)

LEGACY: May be changed to trigger auto-calibration. Need to review system performance.

Manually calibrates the dish motors by “marking” the azimuth/altitude of the current dish position. To configure dish ranges, see motorConfig. To manually move the dish during calibration, see calMove.

Request parameters

Only an upper or lower limit should be sent at once.

Key	Value
<i>altitudeLow</i>	“double” [0.0..90.0] – Set lower angle for altitude, degrees.
<i>altitudeHigh</i>	“double” [0.0..90.0] – Set upper angle for altitude, degrees.
<i>azimuthLow</i>	“double” [0.0..360.0] – Set lower angle for azimuth, degrees.
<i>azimuthHigh</i>	“double” [0.0..360.0] – Set upper angle for azimuth, degrees.

Response parameters

None if a key is provided; response indicates success. Response contains parameters if no keys are sent with the request; parameters include all possible request parameters, but values of null will be present for uncalibrated values.

Errors

If the low and high limits for either azimuth or altitude are equal, the calibration is not valid and will produce an error. This will occur if both upper and lower limits are set in the same request or the dish is not moved between calibration requests.

If a low limit is greater than a high limit, the calibration is not valid and will produce an error.

Example

Request:

```
{ "method": "cal", "params": { "altitudeLow": "4.5" } }
```

Response, success:

```
{ "method": "cal", "params": { TBD } }
```

Response, failure:

```
{ "method": "cal", "params": { TBD }, "error": { "code": -20, "message": "Altitude calibration invalid." }
```

calMove (tentative/legacy)

Move the dish by ticks for manual calibration. Dish will move specified number of ticks from current location; requests should be sent in rapid succession to move the dish continuously.

Request parameters

Only an upper or lower limit should be sent at once.

Key	Value
direction	“left”, “right”, “up”, or “down” – Direction to move.
ticks	Number – Number of ticks to move; defaults to 10.

Response parameters

Response is generated immediately whether notifications are enabled or disabled. Position notifications with azimuth and altitude cannot be generated since calibration is not yet complete.

limit: Boolean, indicates whether dish has reached its limit.

Errors

If the hardware is in an error or safety state, the dish is unable to move and an error will be generated.

capture

Enable or disable Callisto data capture.

Request parameters

Key	Value
mode	"on", "off" – Desired capture state.

Response parameters

Same values possible as request. If the Callisto unit is not functioning properly, an error response describing the cause will be generated.

connect

Initiate connection to controller. Should only be used during troubleshooting.

Request parameters

None.

Response parameters

Successful response indicates controller connection has been established.

Errors

Any error response indicates controller connection was not established. No errors are returned by this command if the controller is in an error state; other commands including “status” should be used to diagnose the controller.

environment

Retrieves latest readings from environmental sensors.

Request parameters

- none -

Response parameters

Key	Value
temp1	“double” – Temperature for sensor 1, degrees C.
	null – Reading not available.
temp2	“double” – Temperature for sensor 2, degrees C.
	null – Reading not available.
humidity1	number [0..100] – Humidity for sensor 1, percentage.
	null – Reading not available.
humidity2	number [0..100] – Humidity for sensor 2, percentage.
	null – Reading not available.
batteryLevel	number [0..100] – Battery level, percentage.
	null – Reading not available.

Errors

Reports an error including affected key names as data and a descriptive message.

Example

Request:

```
{ "method": "environment" }
```

Response, success:

```
{ "method": "environment", "params": { "temp1": "24.0", "temp2": "26.0", "humidity1": 57, "humidity2": 65, "batteryLevel": 100 } }
```

Response, failure:

```
{ "method": "environment", "params": { "temp1": "24.0", "temp2": null, "humidity1": 57, "humidity2": null, "batteryLevel": 100 }, "error": { "code": -10, "message": "sensor2 not detected.", "data": [ "temp2", "humidity2" ] } }
```

environmentConfig

Retrieves and sets environmental constraints. Keys are shared for request and response objects; specifying a key in the request will update the value, and the server will respond with affected keys/values. An empty request will enumerate all current keys/values.

Request parameters

Configured safe ranges are inclusive; a maximum humidity setting of 100% will not trigger shutdown at 100% humidity, and a minimum battery setting of 100% will not trigger shutdown with a full battery.

Key	Possible Values
tempMin1	“double” – Set minimum temperature for sensor 1, degrees C.
	null – Read minimum temperature for sensor 1.
tempMax1	“double” – Set maximum temperature for sensor 1, degrees C.
	null – Read maximum temperature for sensor 1.
tempMin2	“double” – Set minimum temperature for sensor 2, degrees C.
	null – Read minimum temperature for sensor 2.
tempMax2	“double” – Set maximum temperature for sensor 2, degrees C.
	null – Read maximum temperature for sensor 1.
humidityMax1	number [0..100] – Set maximum humidity for sensor 1, percentage.
	null – Read maximum humidity for sensor 1.
humidityMax2	number [0..100] – Set maximum humidity for sensor 2, percentage.
	null – Read maximum humidity for sensor 2.
batteryLevel	number [0..100] – Set minimum battery level, percentage.
	null – Read minimum battery level.

Response parameters

Same types and descriptions as above, but null values are not possible.

Errors

If the system is unable to commit one or more settings due to value being out of range or database error, an error string will be reported including at least one affected key name. No settings will be set if any error occurs while processing the request.

Example

Before the request, tempMax1 and tempMax2 were both configured to "40.0".

Request, valid:

```
{ "method": "environmentConfig", "params": { "tempMax1": "45.0", "tempMax2": null } }
```

Response, success:

```
{ "method": "environmentConfig", "params": { "tempMax1": "45.0", "tempMax2": "40.0" } }
```

Example

Before the request, batteryLevel was configured to 50.

Request, invalid:

```
{ "method": "environmentConfig", "params": { "batteryLevel": 99999 } }
```

Response, failure:

```
{ "method": "environmentConfig", "params": { "batteryLevel": 50 }, "error": { "code": -32602, "message": "Invalid parameter", "data": [ "batteryLevel" ] } }
```

Example

If a request is sent with no parameters, a response containing all parameters will be received.

fitList

Retrieves a list of FIT files and their URLs. If no start and end times are specified, returns last 2 or “limit” FIT files. Sun observation times will only be reported if start and end times are provided.

Request parameters

Key	Possible Values
<i>startTime</i>	“string” – Timestamp of search start time
<i>endTime</i>	“string” – Timestamp of search end time
<i>limit</i>	Number – Maximum number of files to return; default: 2

Response parameters

An array of FIT records will be returned under the “fits” key. Schedule item keys:

Key	Possible Values
name	“string” – Name of FIT file; display name
relUrl	“string” – Relative URL of fit file; download server/relUrl
startTime	“string” – Timestamp of search start time
endTime	“string” – Timestamp of search end time

If no FIT records match the request, the array will be empty. Names of returned items may be used to delete observations (not yet implemented).

Errors

1. If the *startTime* occurs before the *endTime*, an error message indicating an invalid request will be returned.
2. If the system is unable to access its storage subsystem, an error message describing the fault will be returned.

locationConfig

Retrieves and sets location settings. Keys are shared for request and response objects; specifying a key in the request will update the value, and the server will respond with affected keys/values. An empty request will enumerate all current keys/values.

Request parameters

Key	Possible Values
instrument	“string” – Instrument ID string.
	null – Read value.
latitude	“double” [-90.0..+90.0] – Set location latitude, degrees.
	null – Read value.
longitude	“double” [-180.0..+180.0] – Set location longitude, degrees.
	null – Read value.
elevation	“double” – Set elevation above sea level, meters.
	null – Read value.

Response parameters

Same types and descriptions as above, excluding null values.

Errors

If the system is unable to commit one or more settings due to value being out of range or database error, an error string will be reported including at least one affected key name. No settings will be set if any error occurs while processing the request.

mode

Retrieves and sets the system operation mode. The system operation mode will default to “auto” each time the system powers on; the operation mode does not persist across power cycles.

Request parameters

Key	Possible Values
mode	“auto” – System will automatically track sun and follow user observation schedule.
	“maintenance” – System will not follow observation schedule and only respond to immediate position/target commands.

Response parameters

Effective mode will be returned in mode key. No errors possible.

motorConfig

Retrieves and sets motor calibration information. Keys are shared for request and response objects; specifying a key in the request will update the value, and the server will respond with affected keys/values. An empty request will enumerate all current keys/values.

Request parameters

Key	Possible Values
altitudeMaxRuntime	number – Set maximum runtime, milliseconds.
	null – Read maximum runtime.
altitudeDutyCycle	number [0..100] – Set maximum duty cycle, percentage.
	null – Read maximum duty cycle.
altitudeLimitLow	“double” [0.0..90.0] – Set lower limit for altitude, degrees.
	null – Read lower limit for altitude.
altitudeLimitHigh	“double” [0.0..90.0] – Set upper limit for altitude, degrees.
	null – Read upper limit for altitude.
azimuthMaxRuntime	number – Set maximum runtime, milliseconds.
	null – Read maximum runtime.
azimuthDutyCycle	number [0..100] – Set maximum duty cycle, percentage.
	null – Read maximum duty cycle.
azimuthLimitLow	“double” [0.0..360] – Set lower limit for azimuth, degrees.
	null – Read lower limit for azimuth.
azimuthLimitHigh	“double” [0.0..360] – Set upper limit for azimuth, degrees.
	null – Read upper limit for azimuth.

If azimuth low and high limits are equal, the dish will be allowed to rotate freely. Azimuth will “roll over” when passing 0 or 360 degrees.

Response parameters

Same types and descriptions as above, excluding null values.

Errors

If the system is unable to commit one or more settings due to value being out of range or database error, an error string will be reported including at least one affected key name. No settings will be set if any error occurs while processing the request.

parkConfig

Retrieves and sets parking settings. Keys are shared for request and response objects; specifying a key in the request will update the value, and the server will respond with affected keys/values. An empty request will enumerate all current keys/values.

Request parameters

Key	Possible Values
parkAltitude	“double” [0.0..90.0] – Set park position altitude, degrees.
	null – Read park position altitude.
parkAzimuth	“double” [0.0..360] – Set park position azimuth, degrees.
	null – Read park position azimuth.
parkWindyAltitude	“double” [0.0..90.0] – Set windy park position altitude, degrees.
	null – Read windy park position altitude.
parkWindyAzimuth	“double” [0.0..360] – Set windy park position azimuth, degrees.
	null – Read windy park position azimuth.

Response parameters

Same types and descriptions as above, excluding null values.

Errors

If the system is unable to commit one or more settings due to value being out of range or database error, an error string will be reported including at least one affected key name. No settings will be set if any error occurs while processing the request.

position

Retrieves and sets the azimuth and altitude target of the dish. Also reports current actual dish position. Target position will be overridden by scheduled observations unless the system is in maintenance mode.

Request parameters

Key	Possible Values
altitude	“double” [0.0..90.0] – Set target altitude, degrees.
	null – Read target altitude.
azimuth	“double” [0.0..360] – Set target azimuth, degrees.
	null – Read target azimuth.
override	Boolean – This request should override any pending requests.

Response parameters

Key	Possible Values
altitude	“double” [0.0..90.0] – Target altitude, degrees.
	null – No target set.
azimuth	“double” [0.0..360] – Target azimuth, degrees.
	null – No target set.
altitudeCurrent	“double” [0.0..90.0] – Current/actual altitude, degrees.
azimuthCurrent	“double” [0.0..360] – Current/actual azimuth, degrees.

The response will be sent right away if notifications are enabled to indicate the command has been received. The maneuver’s completion will be indicated by a notification.

The response will be sent after the target is reached or the operation fails/times out if notifications are disabled to indicate completion of the maneuver.

In either case, if the maneuver is

Errors

1. If the system is unable to commit one or more settings due to value being out of range or database error, an error string will be reported including at least one affected key name. No settings will be set if any error occurs while processing the request.

2. If the maneuver is aborted by SunTracker override, an error indicating SunTracker override will be generated.

positionConfig (tentative)

Retrieves and sets the azimuth and altitude sensitivity of dish. If actual position falls out of sensitivity margin, corrective action will be made.

Request parameters

Key	Possible Values
altitudeSensitivity	“double” – Set altitude sensitivity, degrees.
	null – Read value.
azimuthSensitivity	“double” – Set azimuth sensitivity, degrees.
	null – Read value.

Values must be greater than or equal to 0.

Response parameters

Same types and descriptions as above, excluding null values.

Errors

If the system is unable to commit one or more settings due to value being out of range or database error, an error string will be reported including at least one affected key name. No settings will be set if any error occurs while processing the request.

scheduleCreate

Retrieves and sets a named target to track, e.g. the sun. The SunTracker driver should be configured with correct location parameters before using the target request.

Request parameters

Key	Possible Values
target	“string” – Name of target to track.
startTime	“string” – Timestamp of search start time
endTime	“string” – Timestamp of search end time

Response parameters

Key	Possible Values
id	Number – Unique schedule item ID.
target	“string” – Name of target to track.
startTime	“string” – Timestamp of search start time
endTime	“string” – Timestamp of search end time

Errors

1. If the scheduled observation conflicts with sun tracking or another observation, the observation cannot be scheduled and an error message indicating this will be returned.
2. If the system does not recognize the target or is otherwise unable to schedule the observation, an error message describing the fault will be returned.

scheduleDelete

Deletes a specific scheduled observation. Users cannot delete sun tracking observations since they are not scheduled or assigned IDs.

Request parameters

Key	Possible Values
Id	Number – ID of schedule item to delete.

Response parameters

None. A successful response indicates that the scheduled observation has been deleted.

Errors

1. If the ID is not recognized, an error will be returned indicating unrecognized ID.
2. If the system is unable to delete the scheduled observation or access its database, an error message describing the fault will be returned.

scheduleList

Retrieves a list of scheduled observations in the specified time window. If no parameters are specified, return all scheduled observations. Sun observation times will only be reported if start and end times are provided.

Request parameters

Key	Possible Values
<i>startTime</i>	“string” – Timestamp of search start time
<i>endTime</i>	“string” – Timestamp of search end time

Response parameters

An array of schedule items will be returned under the “observations” key. Schedule item keys:

Key	Possible Values
id	Number – Unique schedule item ID.
target	“string” – Name of target to track.
startTime	“string” – Timestamp of search start time
endTime	“string” – Timestamp of search end time

If no schedule items match the request, the array will be empty. Unique IDs of returned items may be used to delete observations.

Errors

1. If the *startTime* occurs before the *endTime*, an error message indicating an invalid request will be returned.
2. If the system is unable to access its databases, an error message describing the fault will be returned.

target

Retrieves and sets a named target to immediately track, e.g. the sun. The SunTracker driver should be configured with correct location parameters before using the target request. Manual tracking will be overridden by scheduled tracking unless the system is in maintenance mode.

Request parameters

Key	Possible Values
target	“string” – Name of target to track.
	Boolean – False to clear/disable target tracking.
	null – Read target altitude.
override	Boolean – This request should override observation schedule. MAY BE DEPRECATED.

Response parameters

Key	Possible Values
altitude	“double” [0.0..90.0] – Target altitude, degrees.
	null – No target set.
azimuth	“double” [0.0..360] – Target azimuth, degrees.
	null – No target set.
altitudeCurrent	“double” [0.0..90.0] – Current/actual altitude, degrees.
azimuthCurrent	“double” [0.0..360] – Current/actual azimuth, degrees.

The response will be sent right away if notifications are enabled to indicate the command has been received. The maneuver’s completion will be indicated by a notification.

The response will be sent after the target is reached or the operation fails/times out if notifications are disabled to indicate completion of the maneuver.

In either case, if the maneuver is

Errors

1. If the system is unable to commit one or more settings due to value being out of range or database error, an error string will be reported including at least one affected key name. No settings will be set if any error occurs while processing the request.
2. If the maneuver is aborted by SunTracker override, an error indicating SunTracker override will be generated.

Server Updates

Updates are sent from the server to the client unsolicited. Updates may include information about hardware or system status changes.

position

Reports new/current position of dish to driver. Used to indicate completion or progress of maneuver.

Notification parameters

Key	Possible Values
<i>altitude</i>	“double” [0.0..90.0] – Maneuver’s target altitude, degrees.
<i>azimuth</i>	“double” [0.0..360] – Maneuver’s target azimuth, degrees.
<i>altitudeCurrent</i>	“double” [0.0..90.0] – Current altitude, degrees.
<i>azimuthCurrent</i>	“double” [0.0..360] – Current azimuth, degrees.
<i>completion</i>	Boolean – Present and true if this update indicates that the last maneuver has completed; else absent.
<i>aborted</i>	Boolean – Present and true if this update indicates that the last maneuver has been aborted; else absent.

Altitude, azimuth, and completion are provided for convenience, e.g. showing animations or progress bars. They are always absent if no target is set, e.g. the system is calibrating.