Maestro4 support for ASCOM Telescope

Last revision date: 5 June 2022

Preface

This document describes the mapping of the ASCOM Telescope Interface Standard into the Astrometric Telescope Control System (ATCS) via Maestro. Maestro is ATCS' interface software for Windows-based PCs. For details on the ASCOM Telescope Interface Standard go to the http://ascom-standards.org/ website.

This document addresses the following software versions and/or standards:

- ASCOM Telescope Class version 3.0. Maestro's compliance with this standard is detailed below however please refer to the documentation available from the above website for complete details on the standard.
- Maestro version 4.00.000 (or later).

Maestro can host several ASCOM clients simultaneously. As such it is an ASCOM "hub". The number of clients it can support is limited only by system resources (e.g. memory).

1. Possible Exceptions

The Maestro ASCOM Telescope driver will potentially raise one of the following exceptions for reason/message detailed below:

- NotConnectedException with the message "Not connected to Maestro or Maestro is not connected to the telescope controller.".
- NotConnectedException with the message "Cannot connect to Maestro's client interface. Is Maestro running?".
- NotConnectedException with the message "Maestro is not connected to the telescope controller.".
- DriverException with the message "Error in property/method: <name>. Error message = <message>.". This is a generic error with the message containing more descriptive information.

Additional possible exceptions are listed for each property and method in the following sections.

2. Implementation of ASCOM Telescope Interface Properties

Note: All property values can be read (i.e., have a "get" operation performed on them). Some properties additionally support being written (i.e., "set"). The associated support for each property is detailed below.

2.1. AlignmentMode

The alignment mode of the mount (Alt/Az, Polar, German Polar).

Support when read

ASCOM value	ASCOM description	ATCS meaning
algAltAz	Altitude-Azimuth mount	Settings→For Telescope→Alignment: Alignment Type = AltAz
algGermanPolar	German equatorial mount	Settings→For Telescope→ Alignment: Alignment Type = Polar **and** Settings→For Telescope→Limits & Home Index: Meridian Avoidance = Full(GEM)
algPolar	Polar mount other than German	Settings→For Telescope→ Alignment: Alignment Type = Polar **and** Settings→For Telescope→Limits & Home Index: Meridian Avoidance = None or Lower

2.2. Altitude

The Altitude above the local horizon of the telescope's current position (degrees, positive up).

Support when read

Returns the present ATCS altitude unless ATCS is not aligned in which case 0.0 is returned.

2.3. ApertureArea

The area of the telescope's aperture, taking into account any obstructions (square meters).

Support when read

Not implemented: raises an ASCOM.PropertyNotImplementedException.

2.4. ApertureDiameter

The telescope's effective aperture diameter (meters).

Support when read

Not implemented: raises an ASCOM.PropertyNotImplementedException.

2.5. AtHome

True if the telescope is stopped in the Home position. Set only following a FindHome() operation, and reset with any slew operation.

Support when read

Returns the present ATCS At Home status. This property will read True if the homing operation as successfully completed and False otherwise.

2.6. AtPark

True if the telescope has been put into the parked state by the Park() method. Set False by calling the Unpark() method.

Support when read

Returns True if the telescope has been Parked and has not been moved from the Park position, returns False otherwise. When AtPark is True the following actions, via ASCOM interface, are not allowed: any Slew* method, any Sync* method, AbortSlew method, FindHome method, MoveAxis method, PulseGuide method, or SideOfPier property. UnPark() must be called to get ATCS out of an AtPark state of True. Calling the Park() method while AtPark does nothing.

Note: if AtPark is True then no telescope movement is allowed (other than tracking), via the ASCOM interface, until the Unpark() method is called. This behavior is necessary per the ASCOM Telescope specification however this is not the way that ATCS works in general and if AtPark is True, telescope movement is still allowed via other ATCS interfaces (such as Maestro or handpaddle). Any motion while AtPark is True will immediately set AtPark to False.

2.7. Azimuth

The azimuth at the local horizon of the telescope's current position (degrees, North-referenced, positive East/clockwise).

Support when read

Returns the present ATCS azimuth unless ATCS is not aligned in which case 0.0 is returned.

2.8. CanFindHome

True if this telescope is capable of programmed finding its home position (FindHome() method).

Support when read

If the ATCS home position has been “discovered”, this property reads True, otherwise it reads False.

2.9. CanPark

True if this telescope is capable of programmed parking (using the Park() method).

Support when read

ATCS returns True if its park position is set (and the Park() method will execute without error) and False otherwise.

2.10. CanPulseGuide

True if this telescope is capable of software-pulsed guiding (via the PulseGuide() method).

Support when read

Returns False since the present version off the Maestro Telescope ASCOM driver does not implement PulseGuide().

2.11. CanSetDeclinationRate

True if the DeclinationRate property can be changed to provide offset tracking in the declination axis.

Support when read

ATCS is capable of custom Declination track rate (via DeclinationRate property) and therefore always returns True so long as alignment is complete.

2.12. CanSetGuideRates

True if the guide rate properties used for the PulseGuide() method can be adjusted.

Support when read

Returns False since PulseGuide() moves the telescope at ATCS' current View Velocity and not at the rate given by the corresponding guide rate property.

2.13. CanSetPark

True if this telescope is capable of programmed setting of its park position (SetPark() method).

Support when read

ATCS is capable of setting the park position for the telescope and therefore returns True so long as alignment is complete.

2.14. CanSetPierSide

True if the SideOfPier property can be set, meaning that the mount can be forced to flip.

Support when read

Always returns False. Note: Maestro does support "get SideOfPier". See the SideOfPier property description.

2.15. CanSetRightAscensionRate

True if the RightAscensionRate property can be changed to provide offset tracking in the right ascension axis.

Support when read

ATCS is capable of custom RightAscension track rate (via RightAscensionRate property) and therefore always returns True so long as alignment is complete.

2.16. CanSetTracking

True if the Tracking property can be changed, turning telescope sidereal tracking on and off.

Support when read

ATCS is capable of switching between the current TrackRate and Drift and therefore always returns True so long as alignment is complete.

2.17. CanSlew

True if this telescope is capable of programmed slewing (synchronous or asynchronous) to equatorial coordinates.

Support when read

ATCS is capable of slewing (i.e., GoTo) to celestial and therefore always returns True so long as alignment is complete and Tracking is True.

Note: what is called a "Slew" in ASCOM is called a "GoTo" in ATCS. In ATCS, "Slew" has a different meaning: that is moving the telescope at high speed via a handpaddle/joystick device.

2.18. CanSlewAltAz

True if this telescope is capable of programmed slewing (synchronous or asynchronous) to local horizontal coordinates.

Support when read

ATCS is capable of slewing (i.e. GoTo) to Alt/Az coordinates and therefore always returns True so long as alignment is complete.

2.19. CanSlewAltAzAsync

True if this telescope is capable of programmed asynchronous slewing to local horizontal coordinates.

Support when read

ATCS is capable of asynchronous slewing (i.e. GoTo) to Alt/Az coordinates and therefore always returns True so long as alignment is complete.

2.20. CanSlewAsync

True if this telescope is capable of programmed asynchronous slewing to equatorial coordinates.

Support when read

ATCS is capable of asynchronous slewing (i.e. GoTo) to celestial coordinates and therefore always returns True so long as alignment is complete.

2.21. CanSync

True if this telescope is capable of programmed synching to equatorial coordinates.

Support when read

ATCS is always capable of sync and thus returns True.

Note: see details under the SyncToCoordinates and SyncToTarget methods on differences of how ATCS handles syncing pre and post alignment.

2.22. CanSyncAltAz

True if this telescope is capable of programmed synching to local horizontal coordinates

Support when read

ATCS is always capable of sync and thus returns True.

Note: see details under the SyncToAltAz method on differences of how ATCS handles syncing pre and post alignment.

2.23. CanUnpark

True if this telescope is capable of programmed unparking (Unpark() method).

Support when read

ATCS returns True if its park position is set (the only necessary criteria for the UnPark() method to execute without error) and False otherwise.

2.24. Connected

Set True to connect to the device hardware. Set False to disconnect from the device hardware. You can also read the property to check whether it is connected. This reports the current hardware state.

Support when read

Maestro returns True if it is connected to ATCS and False otherwise.

Support when written

Maestro is an ASCOM "local server" and, as such, can connect to multiple hardware devices and, for each device, allows possibly multiple ASCOM clients simultaneous access. Therefore, in Maestro, once the communications port is selected Maestro will then continuously attempt to connect to the hardware to assure hardware access is available for all possible clients and the Maestro user interface. As a result, hardware connection/disconnection is not initiated from the ASCOM client and writing of this “Connected” property does nothing other than raise an error if the client is attempting to set Connected to true but Maestro is not itself connected to the hardware.

Additionally, an ASCOM client cannot disconnect Maestro from the hardware. Therefore, writing of this “Connected” property to false does nothing.

2.25. Declination

The declination (degrees) of the telescope's current equatorial coordinates, in the coordinate system given by the EquatorialSystem property. Reading the property will raise an error if the value is unavailable.

Support when read

Returns the present ATCS declination unless ATCS is not aligned in which case 0.0 is returned.

Note: Maestro’s ASCOM Telescope driver only uses EquatorialCoordinateType.equTopocentric. Declination is thus reported in this Epoch.

2.26. DeclinationRate

The declination tracking rate (arcseconds per SI second, default = 0.0).

Support when read

The value of ATCS’ Custom TrackRate (offset from Sidereal) is reported in units of arcseconds per second. Note: ATCS uses units of arcseconds per standard hour therefore ATCS’ Custom TrackRate in Dec (R) is converted per the following sequence:

Dec arcseconds per second = R ÷ 3600 seconds per hour

Support when written

When this property is written two things happen:

1. The property value, which (per ASCOM spec) is in arcseconds per second, is converted to ATCS’ required units of arcseconds per standard hour (per the reverse of the above sequence) and ATCS’ Custom Track Rate in Dec is set accordingly.
2. If the property value is set to a non-zero value, and ATCS is aligned, then ATCS’ Track Rate is set to Custom.

DeclinationRate is bounded to ±7200.0 arcseconds per second (±2 degrees per second).

Note: ASCOM defines DeclinationRate as being applied as an offset to the presently selected track rate. ATCS does not work this way. Rather, if either DeclinationRate or RightAscensionRate are non-zero then ATCS’ TrackRate is set to Custom. If they are both zero then:

- If ATCS’ TrackRate is Custom it is set to Sidereal.
- If ATCS’ TrackRate is NOT Custom nothing is done.

2.27. Description

Returns a description of the device, such as manufacturer and model number.

Support when read

Maestro reports: “Astrometric telescope control via Maestro”.

Note: this is the text listed in the driver selection “drop down” in the ASCOM driver chooser.

2.28. DoesRefraction

True if the telescope or driver applies atmospheric refraction to coordinates.

Support when read

Maestro reports: “Astrometric telescope control via Maestro”.

Reports (True or False) if ATCS’ refraction correction is enabled or not. If refraction correction is enabled, celestial coordinates input to ATCS (including through the ASCOM interface) are forward-corrected for refraction to arrive at observed target coordinates and coordinates output by ATCS (i.e., achieved coordinates) are backward corrected for refraction to provide apparent coordinates.

Note: Altitude and Azimuth are taken as observed coordinates all of the time regardless of the value of DoesRefraction.

Support when written

Directly sets ATCS’ refraction correction to True or False (i.e. enabled or disabled).

2.29. DriverInfo

Descriptive and version information about this ASCOM driver.

Support when read

Maestro reports: “ASCOM driver: Astrometric telescope control via Maestro - v.M.N.R. Copyright Astrometric Instruments, Inc.”. M.N.R is major version, minor version and revision.

2.30. DriverVersion

A string containing only the major and minor version of the driver.

Support when read

Maestro reports: “M.N.R” which is major version, minor version and revision.

2.31. EquatorialSystem

Equatorial coordinate system used by this telescope (e.g. Topocentric or J2000).

Support when read

Maestro reports: EquatorialCoordinateType.equTopocentric indicating that coordinates input to or read from this driver are for the current date having allowed for annual aberration, precession and nutation.

Note: ATCS JNow coordinates are what ASCOM calls Apparent coordinates. ATCS does not correct for the sub-arcsec diurnal aberration necessary to get from what ASCOM calls Apparent to what ASCOM calls Topocentric. However, this is only sub-arcsec.

Note: ATCS only provides JNow/Apparent for celestial coordinates “back out” of the hardware. So, a decision had to be made when implementing this driver:

- JNow/Apparent could be converted back to J<other> using the ASCOM.Astrometry.Transform object however this object does not accept J2050. Additionally, access to each of the equatorial properties (i.e., Declination and RightAscension) would require running the transform and wasting one half of the output (i.e., waste Dec result when after RA, etc.).
- We therefore just report that this driver only supports JNow (EquatorialCoordinateType.equTopocentric) and avoid the lack of support for J2050 in ASCOM.Astrometry.Transform.

2.32. FocalLength

The telescope's focal length, meters.

Support when read

Not implemented: raises an ASCOM.PropertyNotImplementedException.

2.33. GuideRateDeclination

The current Declination movement rate offset for telescope guiding (degrees/sec).

Support when read

To-be implemented but presently not implemented: raises an ASCOM.PropertyNotImplementedException.

Support when write

To-be implemented but presently not implemented: raises an ASCOM.PropertyNotImplementedException.

2.34. GuideRateRightAscension

The current Right Ascension movement rate offset for telescope guiding (degrees/sec).

Support when read

To-be implemented but presently not implemented: raises an ASCOM.PropertyNotImplementedException.

Support when write

To-be implemented but presently not implemented: raises an ASCOM.PropertyNotImplementedException.

2.35. InterfaceVersion

The interface version number that this device supports.

Support when read

This driver implements the ASCOM ITelescopeV3 Interface and thus returns 3.

2.36. Name

The short name of the driver, for display purposes.

Support when read

Maestro reports: “MaestroTelescope”.

2.37. RightAscension

The right ascension (hours) of the telescope's current equatorial coordinates, in the coordinate system given by the EquatorialSystem property.

Support when read

Returns the present ATCS right ascension unless ATCS is not aligned in which case 0.0 is returned.

Note: Maestro’s ASCOM Telescope driver only uses EquatorialCoordinateType.equTopocentric. Right ascension is thus reported in this Epoch.

2.38. RightAscensionRate

The right ascension tracking rate offset from sidereal (seconds per sidereal second, default = 0.0).

Support when read

The value of ATCS’ Custom TrackRate (offset from Sidereal) is reported in units of seconds per sidereal second. Note: ATCS uses units of arcseconds per standard hour therefore ATCS’ Custom TrackRate in RA (R) is converted per the following sequence:

RA seconds per hour = R ÷ 15 arcseconds per RA second

RA seconds per second = RA seconds per hour ÷ 3600 seconds per hour

RA seconds per sidereal second = RA seconds per second × 0.9972695677 seconds per sidereal second

Support when written

When this property is written two things happen:

1. The property value, which (per ASCOM spec) is in seconds per sidereal second, is converted to ATCS' required units of arcseconds per standard hour (per the reverse of the above sequence) and ATCS' Custom Track Rate in Dec is set accordingly.
2. If the property value is set to a non-zero value, and ATCS is aligned, then ATCS' Track Rate is set to Custom.

RightAscensionRate is bounded to ±478.68 seconds per sidereal second (±2 degrees per second).

Note: ASCOM defines RightAscensionRate as being applied as an offset to the presently selected track rate. ATCS does not work this way. Rather, if either DeclinationRate or RightAscensionRate are non-zero then ATCS' TrackRate is set to Custom. If they are both zero then:

- ♦ If ATCS' TrackRate is Custom it is set to Sidereal.
- ♦ If ATCS' TrackRate is NOT Custom nothing is done.

2.39. SideOfPier

Indicates the pointing state of the mount.

Support when read

If ATCS is not aligned then returns PierSide.pierUnknown. Otherwise, the behavior depends on ATCS' AlignmentMode:

- If AlignmentMode = algGermanPolar: Returns PierSide .pierEast or PierSide.pierWest.
- If AlignmentMode != algGermanPolar: Returns PierSide.pierUnknown.

Note: ASCOM nomenclature uses side of pier that the telescope is on. ATCS nomenclature uses side of the sky that the telescope is configured to point into (i.e. East or West). These are opposite. The value of this property is reported in ASCOM terms to be compliant with the spec.

Support when written

Not implemented: raises an ASCOM.PropertyNotImplementedException.

2.40. SiderealTime

The local apparent sidereal time from the telescope's internal clock (hours, sidereal).

Support when read

Fully supported.

2.41. SiteElevation

The elevation above mean sea level (meters) of the site at which the telescope is located.

Support when read

Not implemented: raises an ASCOM.PropertyNotImplementedException.

2.42. SiteLatitude

The geodetic(map) latitude (degrees, positive North, WGS84) of the site at which the telescope is located.

Support when read

Fully supported.

Support when written

Fully supported.

Note: ATCS supports 4 sites. Setting reding/writing SiteLatitude works on the presently selected site.

Note: writing this property will void alignment.

2.43. SiteLongitude

The longitude (degrees, positive East, WGS84) of the site at which the telescope is located.

Support when read

Fully supported.

Support when written

Fully supported.

Note: ATCS supports 4 sites. Setting reding/writing SiteLongitude works on the presently selected site.

Note: writing this property will void alignment.

2.44. Slewing

True if telescope is currently moving in response to one of the Slew methods or the MoveAxis(TelescopeAxes, Double) method, False at all other times.

Support when read

Returns True if ATCS' MoveMode does not contain the work "Track" (e.g. "Track/Sidereal"). Returns False otherwise.

Note: The desertion of the Slewing property will be extended beyond the time when ATCS MoveMode returns to "Track" by the value of the SlewSettleTime property.

Note: Slewing can be set for a short period even if a Slew does not occur. This is because it can take some time to fully vet the legality of the slew.

2.45. SlewSettleTime

Specifies a post-slew settling time (sec.).

Synchronous slewing methods will not return, and the Slewing property will not become False, until the slew completes and the SlewSettleTime has elapsed.

Support when read

Fully supported.

Support when written

Fully supported.

2.46. SupportedActions

Returns the list of custom action names supported by this driver.

Support when read

Per the ASCOM specification, this property should return the device-specific actions that the driver supports. This driver does not support any device-specific actions and returns a blank list.

Note: ATCL commands can be passed directly to the hardware using the CommandString() method.

2.47. TargetDeclination

The declination (degrees, positive North) for the target of an equatorial slew or sync operation

Support when read

Fully supported.

Support when written

Fully supported.

Note: TargetDeclination is assumed to be valid for EquatorialCoordinateType.equTopocentric.

2.48. TargetRightAscension

The right ascension (hours) for the target of an equatorial slew or sync operation

Support when read

Fully supported.

Support when written

Fully supported.

Note: TargetRightAscension is assumed to be valid for EquatorialCoordinateType.equTopocentric.

2.49. Tracking

The state of the telescope's sidereal tracking drive.

Support when read

ASCOM value	ATCS meaning
False	Not tracking (i.e. in a TrackRate of Drift)
True	Yes tracking but not necessarily at Sidereal TrackRate (e.g. will be at Custom TrackRate if either DeclinationRate or RightAscensionRate are non-zero or could be tracking at Lunar, Solar, etc.)

Support when written

ASCOM value	ATCS meaning
False	Causes ATCS to enter a TrackRate of Drift and save the old TrackRate.
True	Causes ATCS to change the TrackRate back to the old saved TrackRate. If ATCS powered-up in TrackRate of Drift then setting this property to True will set the TrackRate to Sidereal.

2.50. TrackingRate

The current tracking rate of the telescope's sidereal drive.

Support when read

Provides the present TrackRate within ATCS:

ATCS is presently tracking at	ASCOM value
Sidereal rate	driveSidereal
Lunar rate	driveLunar
Solar rate	driveSolar
Custom rate	driveSidereal

Note: ATCS does not have a separate King (ASCOM driveKing) tracking rate. Instead, if refraction correction is enabled, ATCS' tracking is continuously corrected for the effects of atmospheric refraction regardless of its TrackRate. The ASCOM Client can enable/disable this ability through the DoesRefraction property.

Note: if the property reads driveSidereal, and it is necessary to determine if ATCS is using its Custom TrackRate, then the RightAscension rate and DeclinationRate should be polled… if either are non-zero then ATCS is using its Custom TrackRate.

Note: if ATCS' TrackRate is Drift then this property will report driveSidereal however the Tracking property will return False. ASCOM does not use ATCS' concept that Drift is just another TrackRate.

Support when written

When this property is written to one of the ASCOM values in the above table ATCS' TrackRate will change accordingly.

2.51. TrackingRates

Returns a collection of supported DriveRates values that describe the permissible values of the TrackingRate property for this telescope type.

Support when read

Returns a collection of supported tracking rate values for ATCS. The collection includes three values: driveSidereal, driveLunar and driveSolar.

Support when written

Not a writeable property.

2.52. UTCDate

The UTC date/time of the telescope's internal clock.

Support when read

Returns the UTC date/time read from ATCS for the presently selected Site.

Support when written

Not implemented: raises an ASCOM.PropertyNotImplementedException.

3. Implementation of ASCOM Telescope Interface Methods

3.1. AbortSlew()

Stops a slew in progress.

Fully compliant.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: this method aborts all non-tracking motion.

3.2. Action()

Invokes the specified device-specific custom action.

This driver does not support any device-specific actions and hence raises an ASCOM. MethodNotImplementedException.

Note: ATCL commands can be passed directly to the hardware using the CommandString() method.

3.3. AxisRates()

Determine the rates at which the telescope may be moved about the specified axis by the MoveAxis(TelescopeAxes, Double) method.

For reasons described in the MoveAxis() method this is not implemented and simply returns an empty collection per the ASCOM specification.

3.4. CanMoveAxis(TelescopeAxes Axis)

True if this telescope can move the requested axis.

For reasons described in the MoveAxis() method this is not implemented and returns False for each valid value of Axis.

3.5. CommandBlind(string Command, bool Raw = false)

Transmits an arbitrary string to the device and does not wait for a response. Optionally, protocol framing characters may be added to the string before transmission.

Not supported and raises an ASCOM. MethodNotImplementedException.

3.6. CommandBool(string Command, bool Raw = false)

Transmits an arbitrary string to the device and waits for a boolean response. Optionally, protocol framing characters may be added to the string before transmission.

Not supported and raises an ASCOM. MethodNotImplementedException.

3.7. CommandString(string Command, bool Raw = false)

Transmits an arbitrary string to the device and waits for a string response. Optionally, protocol framing characters may be added to the string before transmission.

Allows the Client to send ATCL string commands directly to ATCS. ATCS' response to the ATCL command is provided in the response string.

Note: ATCS' primary interface protocol is called the Astrometric Telescope Control Language (ATCL). ATCL is typically communicated over USB or Ethernet to ATCS' Client interface port (labeled "Com"). ATCL provides a very rich syntax for telescope control. Maestro controls ATCS through ATCL and the CommandString() method allows ASCOM Clients direct access to the ATCL interface to ATCS.

To pass an ATCL command directly to ATCS, call the CommandString() method with a string parameter that is an ATCL command. The response string of the CommandString() function is the response from ATCS. For example, to set ATCS' ViewVel4 to 0.8deg/sec you could use the following VBA CommandString() code:

```
Set Scope = CreateObject("ASCOM.MaestroTelescope.Telescope")
Scope.Connected = true
Response = Scope.CommandString( "KSv40.8deg/sec" )
```

For more information on ATCL contact Astrometric at support@astrometric.com.

Note: the Raw parameter is ignored however the behavior of this command is that the Command string is always sent to ATCS, and the return string provided back to the Client, verbatim (i.e. without processing or additional delimiters).

Additional notes:

- Per ATCL protocol, commands must be proceeded by '!' and terminated with ';'. The CommandString() method does not require these characters be included in the string (they are included by Maestro).
- Per ATCL protocol, all strings returned from the hardware are terminated with a semicolon (';'). The CommandString() method removes this semicolon from return strings.
- Not all ATCL commands return a string. Many return only the ATCL_ACK character (hex code 0x8F) to indicate all is OK, or the ATCL_NACK character (hex code 0xA5) to indicate a problem. When there is no return string from the ATCL command, CommandString() will return an empty string.
- If the ATCL command does not return a string or it does not return ATCL_ACK then there is a problem with the ATCL command and an ASCOM. InvalidValueException is raised.

3.8. DestinationSideOfPier(double RightAscension, double Declination)

Predict side of pier for German equatorial mounts.

Not supported and raises an ASCOM. MethodNotImplementedException.

3.9. FindHome()

Locates the telescope's "home" position (synchronous).

Fully compliant. This method does not return until the home position has been found (i.e., it is "synchronous") and alignment is completed from the pre-discovered home position.

Note: this method requires that the ATCS home index switches have been "discovered" on both axes.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

3.10. MoveAxis(TelescopeAxes Axis, double Rate)

Move the telescope in one axis at the given rate.

This method is not implemented since the mapping of a collection or Rate objects to ATCS' 4 ViewVels and Slew (for 5 total rates) is possible but problematic since ViewVels and Slew (MaxVel) settings can change rendering any collection of rates (as provided by AxisRates() method) potentially stale. This becomes non-tractable. Calling this method will raise an ASCOM. MethodNotImplementedException.

3.11. Park()

Move the telescope to its park position, stop all motion, and set AtPark to True (asynchronous).

Fully compliant. This method starts an ATCS GoToPark command and returns (i.e., it is "asynchronous"). The client must poll the AtPark property to determine when the GoToPark is complete.

Once parked:

- ♦ The AtPark property registers True and no motion is allowed (via ASCOM Client request) until the UnPark() method is called.
- ♦ Alignment is voided.

Note: If the park position has not been "marked" (i.e., set) in ATCS then this method will raise an ASCOM.InvalidOperationException with the message "Cannot Park the telescope until the Parked position is marked.". The park position can be marked using the SetPark() method.

Park works even when not aligned since park position is recorded as an ATCS axial position the GoTo of which does not require alignment.

See additional notes under the AtPark property description.

3.12. PulseGuide(GuideDirections Direction, int Duration)

Moves the scope in the given direction for the given interval or time at the rate given by the corresponding guide rate property.

Not supported and raises an ASCOM. MethodNotImplementedException.

Note: per the ASCOM specification, PulseGuide() does not set the Slewing property.

3.13. SetPark()

Sets the telescope's park position to be its current position.

Fully compliant. ATCS saves the present Axial position as the park position.

3.14. SetupDialog()

Launches a configuration dialog box for the driver. The call will not return until the user clicks OK or cancel manually.

Note: There are no settings to be made from this dialog. It simply displays the text "Maestro telescope settings are made directly from the

Maestro program. Click OK to continue."

3.15. SlewToAltAz(double Azimuth, double Altitude)

Move the telescope to the given local horizontal coordinates, return when slew is complete (synchronous).

Fully supported.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is True, calling this method will raise and ASCOM.InvalidOperationException.

3.16. SlewToAltAzAsync(double Azimuth, double Altitude)

Start a move of the telescope to the given local horizontal coordinates and return immediately (asynchronous).

Fully supported.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is True, calling this method will raise and ASCOM.InvalidOperationException.

3.17. SlewToCoordinates(double RightAscension, double Declination)

Move the telescope to the given equatorial coordinates, return when slew is complete (synchronous).

Fully supported.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is False, calling this method will raise and ASCOM.InvalidOperationException.

3.18. SlewToCoordinatesAsync(double RightAscension, double Declination)

Move the telescope to the given equatorial coordinates, return immediately after starting the slew (asynchronous).

Fully supported.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is False, calling this method will raise and ASCOM.InvalidOperationException.

3.19. SlewToTarget()

Move the telescope to the TargetRightAscension and TargetDeclination coordinates, return when slew complete (synchronous).

Fully supported.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is False, calling this method will raise and ASCOM.InvalidOperationException.

3.20. SlewToTargetAsync()

Move the telescope to the TargetRightAscension and TargetDeclination coordinates, returns immediately after starting the slew (asynchronous).

Fully supported.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is False, calling this method will raise and ASCOM.InvalidOperationException.

3.21. SyncToAltAz(double Azimuth, double Altitude)

Matches the scope's local horizontal coordinates to the given local horizontal coordinates.

Fully supported.

Note: If a “Synch” command is received when ATCS is not yet aligned then the target coordinates are taken as alignment coordinates for a one-star alignment if one-star alignment is presently allowed. This is a convenience and not part of the ASCOM spec.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is True, calling this method will raise and ASCOM.InvalidOperationException.

3.22. SyncToCoordinates(double RightAscension, double Declination)

Matches the scope's equatorial coordinates to the given equatorial coordinates.

Fully supported.

Note: If a “Synch” command is received when ATCS is not yet aligned then the target coordinates are taken as alignment coordinates for a one-star alignment if one-star alignment is presently allowed. This is a convenience and not part of the ASCOM spec.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is False, calling this method will raise and ASCOM.InvalidOperationException.

3.23. SyncToTarget()

Matches the scope's equatorial coordinates to the given equatorial coordinates.

Fully supported.

Note: If a “Synch” command is received when ATCS is not yet aligned then the target coordinates are taken as alignment coordinates for a one-star alignment if one-star alignment is presently allowed. This is a convenience and not part of the ASCOM spec.

Note: if the AtPark property is True, calling this method will raise an ASCOM. ParkedException.

Note: per the ASCOM specification, if the Tracking property is False, calling this method will raise and ASCOM.InvalidOperationException.

3.24. Unpark()

Takes telescope out of the Parked state.

Fully supported however Maestro behaves differently with regard to the ASCOM spec requirement that Unpark() does nothing if AtPark is False in that regardless of the state of AtPark Maestro proceeds as follows:

- If ATCS is not aligned, Maestro submits an AlignFromLast command to ATCS to assure that ATCS is aligned after the Unpark().
- Maestro turns on ATCS’s tracking.

This behavior is necessary to assure that ATCS is aligned after Unpark() even if movement outside of ASCOM caused AtPark to be False (see note below...)

Note: the ASCOM spec states that the value of Tracking is unknown after Unpark however ATCS starts tracking after Unpark().

See additional notes under the AtPark property description.