

Maestro support for ASCOM

Copyright 2005-2013 Astrometric Instruments, Inc.

Last revision date: 15-Jan-2013

This document can be ordered as Astrometric Instruments' part DOC-08A

Preface

This document describes the mapping of the ASCOM Telescope Interface Standard into the Astrometric Telescope Control System (ATCS). ASCOM functionality is provided through Maestro software. Maestro is SkyWalker's interface software for Windows-based PCs. SkyWalker is the telescope motor and accessory controller, and hub, of ATCS.

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 Interface Standard version 2.0. Maestro's compliance with this standard is detailed below however please refer to the document titled "Telescope Interface Standard 2.0 -- Programmer's Reference" (available from the above website) for complete details on the standard.
- Maestro version 1.00.000 (or later)

Note: Current ATCS support for ASCOM is derived from original support (released July 2002) provided in older SkyGuide software (now superseded by Maestro). SkyGuide only supported the ASCOM Telescope Interface Standard 1.0.

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

Sections of this document

- Possible Trappable Errors
- Implementation of ASCOM Telescope Interface Properties
- Implementation of ASCOM Telescope Interface Methods
- Using SkyWalker/Maestro Remotely with ASCOM
- SkyWalker's Asynchronous Message Reporting to ASCOM Client Applications
- More Resources

Possible Trappable Errors

The Maestro ASCOM Telescope Interface driver will potentially raise one of these trappable error codes:

- **0x80040400:** For a un-implemented ASCOM property or method: 0x80040400 and the message: "<name> property or method is not implemented in Maestro's ASCOM driver."
- **0x80040401:** If Maestro is not found (i.e. Maestro not running): 0x80040401 and the message: "Maestro not found".
- **0x80040402:** If SkyWalker is not found (i.e. SkyWalker is not running or not connected to Maestro): 0x80040402 and the message: "SkyWalker not found".
- **0x80040404:** For a range error in a property value or method parameter: 0x80040404 and the message: "<name> ASCOM property or method is used with a value out of range.". This error will occur when, for example, attempting to set the TargetDeclination property outside of the range +90 to -90.

Note: it is possible that this error is not raised and instead a warning, indicating the invalidity in the parameter, is provided from SkyWalker through Maestro (see description of error code 0x80040411 below).

Maestro support for ASCOM

- **0x80040405:** If a property within the ASCOM driver is not properly set: 0x80040405 and the message “<name> ASCOM property has not yet been set.”
- **0x80040410:** For error/warning returned back from Maestro (i.e. not from SkyWalker): 0x80040410 and the message defined in the documentation that follows. One example is “King rate unnecessary. SkyWalker provides continuous refraction correction. No King rate support.”
- **0x80040411:** For messages (i.e. status, warnings, alerts and fatal errors) generated asynchronously in SkyWalker (and raised through Maestro): 0x80040411 and SkyWalker’s message text. Note: the message level must be set as described in the section “SkyWalker’s Asynchronous message reporting to ASCOM client applications”.

Note: the source of messages associated with reporting problems in setting properties or executing methods is largely SkyWalker (and not Maestro). Maestro forwards them to the ASCOM Client by raising them as a trappable error with the code 0x80040411. This is an important feature since it allows Maestro to be “light weight” and do little of its own message generation and also provides one source for legality checking (i.e. SkyWalker). In its roll as an ASCOM Telescope Server, Maestro is simply a shell interface to SkyWalker.

Implementation of ASCOM Telescope Interface Properties

AlignmentMode: The alignment mode of the mount.

Support when read

ASCOM value	ASCOM description	SkyWalker meaning
algAltAz	Altitude-Azimuth mount	Settings/AlignmentSetup/AlignmentType = AltAz or NearlyAltAz or Arbitrary
algGermanPolar	German equatorial mount	Settings/AlignmentSetup/AlignmentType = Polar or NearlyPolar and Settings/Instrumental/MotionLimits/MeridianLimits = Full(GEM)
algPolar	Polar mount other than German	Settings/AlignmentSetup/AlignmentType = Polar or NearlyPolar and Settings/Instrumental/MotionLimits/MeridianLimits = None or Lower

Note: if Settings/AlignmentSetup/AlignmentType is Permanent then AlignmentMode is the same as the AlignmentMode prior to the Settings/AlignmentSetup/AlignmentType being set to Permanent.

Support when written

Not a writeable property.

Altitude: The observed Altitude (degrees, positive up) that the telescope is currently pointing.

Support when read

Fully supported. If SkyWalker is not aligned then 0.0 is return.

Support when written

Maestro support for ASCOM

Not a writeable property.

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

Support when read

Not supported.

Support when written

Not a writeable property.

ApertureDiameter: The telescope's effective aperture diameter (meters).

Support when read

Not supported.

Support when written

Not a writeable property.

AtHome: Indicates if the telescope has stopped at the Home position.

Support when read

If a Home Index Sensor (and Automation Module support) is installed: this property reads True if the Telescope has been homed. Set only following a FindHome() operation, and reset with any slew operation.

If a Home Index Sensor is **not** installed: this property always reads False (per ASCOM Telescope V2.0 specification). Note: SkyWalker “At Home” indicates ONLY that the telescope is physically at the home switches. ASCOM AtHome means that it is at the home switches AND it is aligned/ready-to-go. Hence, if SkyWalker is not aligned then this property will return False even if the telescope is “At Home” in SkyWalker.

Support when written

Not a writeable property.

AtPark: Indicates if the telescope has been Parked and has not been moved from the Park position.

Support when read

True if the telescope has been Parked and has not been moved from the Park position, 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 SkyWalker out of an AtPark state of True. Calling the Park() method while AtPark does nothing.

Support when written

Not a writeable property.

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 V2.0 specification however this is **not** the way that ATCS works in general and if AtPark is True, telescope movement **is** still allowed via other SkyWalker interfaces (such as Maestro or HP2). Any motion while AtPark is True will immediately set AtPark to False.

Maestro support for ASCOM

Azimuth: The observed Azimuth (degrees, 0.0 North, positive Eastward) that the telescope is currently pointing.

Support when read

Fully supported. If SkyWalker is not aligned then 0.0 is return.

Support when written

Not a writeable property.

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

Support when read

If a Home Index Sensor (and Automation Module support) is installed, and the home position has been “discovered”, this property reads True, otherwise it reads False.

Support when written

Not a writeable property.

CanPark: True if this telescope is capable of parking (via the Park() method).

Support when read

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

Support when written

Not a writeable property.

CanPulseGuide: True if this telescope is capable of programmed guiding (PulseGuide() method).

Support when read

Fully supported. SkyWalker is capable of “pulse guiding” and therefore always returns True.

Support when written

Not a writeable property.

CanSetDeclinationRate: True if the DeclinationRate property can be changed to provide offset tracking in the Declination axis.

Support when read

Fully supported. SkyWalker is capable of custom Declination track rate (via DeclinationRate property) and therefore always returns True.

Support when written

Not a writeable property.

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

Support when read

Maestro support for ASCOM

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

Support when written

Not a writeable property.

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

Support when read

Fully supported. SkyWalker is capable of setting the park position for the telescope and therefore always returns True.

Support when written

Not a writeable property.

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

Support when read

Not supported. Maestro always returns False. Note: Maestro does support "get SideOfPier". See the SideOfPier property description.

Support when written

Not a writeable property.

CanSetRightAscensionRate: True if the RightAscensionRate property can be changed to provide offset tracking in the Right Ascension axis.

Support when read

Fully supported. SkyWalker is capable of custom Right Ascension track rate (via RightAscensionRate property) and therefore always returns True.

Support when written

Not a writeable property.

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

Support when read

Fully supported. SkyWalker is capable of switching between the current TrackRate and Drift and therefore always returns True.

Support when written

Not a writeable property.

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

Support when read

Fully supported. SkyWalker is capable of slewing (i.e. GoTo) to celestial and therefore always returns True.

Maestro support for ASCOM

Support when written

Not a writeable property.

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

CanSlewAltAz: True if this telescope is capable of programmed slewing (synchronous or asynchronous) to Alt/Az coordinates.

Support when read

Fully supported. SkyWalker is capable of slewing (i.e. GoTo) to Alt/Az coordinates and therefore always returns True.

Support when written

Not a writeable property.

CanSlewAltAzAsync: True if this telescope is capable of programmed asynchronous slewing to Alt/Az coordinates.

Support when read

Fully supported. SkyWalker is capable of asynchronous slewing (i.e. GoTo) to Alt/Az coordinates and therefore always returns True.

Support when written

Not a writeable property.

CanSlewAsync: True if this telescope is capable of programmed asynchronous slewing to celestial coordinates.

Support when read

Fully supported. SkyWalker is capable of asynchronous slewing (i.e. GoTo) to celestial coordinates and therefore always returns True.

Support when written

Not a writeable property.

CanSync: True if this telescope is capable of programmed synching to celestial coordinates.

Support when read

Fully supported. SkyWalker is capable of this and therefore always returns True.

Support when written

Not a writeable property.

Note: what ASCOM calls “Synching” SkyWalker calls “Calibration”.

CanSyncAltAz: True if this telescope is capable of programmed synching to Alt/Az coordinates.

Support when read

Fully supported. SkyWalker is capable of this and therefore always returns True.

Support when written

Maestro support for ASCOM

Not a writeable property.

CanUnpark: True if this telescope is capable of programmed un-parking (Unpark() method).

Support when read

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

Support when written

Not a writeable property.

Connected: True if telescope connected, False otherwise.

Support when read

- Fully supported. Maestro returns True if it is connected to SkyWalker and False otherwise.

Note: if any property/method is accessed when not connected to SkyWalker then the error 0x80040402 is raised with the message: "SkyWalker not found".

Support when written

If Maestro is running, it continuously attempts to connect to SkyWalker. The Client can write Connected however Maestro ignores the action since it will have already connected to SkyWalker if possible.

If Connected is written, and Maestro has not connected to SkyWalker (for whatever reason) then the trappable error code 0x80040402 will be raised with the message "SkyWalker not found".

Declination: The declination (degrees) of the telescope's current local equatorial coordinates.

Support when read

Fully supported. If SkyWalker is not aligned then 0.0 is return.

Support when written

Not a writeable property.

Note: Declination is provided in the Apparent Declination form. This is in the form of "local topocentric coordinate system" in ASCOM terms (see more information under the EquatorialSystem property). SkyWalker does not "backwards" correct Apparent celestial coordinates to standard reference frame (i.e. so called "catalog" coordinates in SkyWalker).

DeclinationRate: The declination tracking rate (arcseconds per second).

Support when read

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

Dec arcseconds per second = $R \div 3600$ seconds per hour

Support when written

When this property is written two things happen:

Maestro support for ASCOM

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

DeclinationRate is bounded to ± 277.77 arcseconds per second.

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

- If SkyWalker's TrackRate is Custom it is set to Sidereal.
- If SkyWalker's TrackRate is NOT Custom nothing is done.

Note: SkyWalker's Custom TrackRate is the same as Sidereal prior to SkyWalker being aligned.

Description: The long description of the telescope.

Support when read

Maestro reports:

“Unknown telescope controlled by Astrometric Instruments Telescope Control system (i.e. Maestro controlling SkyWalker)”.

Support when written

Not a writeable property.

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

Support when read

Reports (True or False) if SkyWalker's refraction correction is enabled or not. If refraction correction is enabled, celestial coordinates input to SkyWalker (including through the ASCOM interface) are forward-corrected for refraction to arrive at observed target coordinates and coordinates output by SkyWalker (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 SkyWalker's refraction correction to True or False (i.e. enabled or disabled).

DriverInfo: Descriptive and version information about this ASCOM Telescope driver.

Support when read

Maestro reports:

“Maestro ASCOM Telescope driver.

Driver Version = x.x.x

Astrometric Instruments, Inc.

Copyright © 2005, Astrometric Instruments, Inc.

ASCOM interface to SkyWalker (through Maestro). Visit www.astrometric.com for more info.”

Maestro support for ASCOM

Support when written

Not a writeable property.

DriverVersion: Provides the major and minor version of this ASCOM Telescope driver.

Support when read

Driver version in the form m.n, where ‘m’ is major version and ‘n’ is minor version.

Support when written

Not a writeable property.

EquatorialSystem: Equatorial coordinate system used by SkyWalker.

Support when read

Indicates the Epoch of Entry used by SkyWalker. This specifies the coordinate system that SkyWalker allows input coordinates (including coordinates from ASCOM clients) to be assumed valid. SkyWalker supports coordinates in the ICRS reference frame for years 2000-2050 and also supports the local topocentric reference frame (i.e. Epoch of “now”). This allows for the following ASCOM EquatorialCoordinateType values:

Constant	Value	Description
equLocalTopocentric	1	Reported when SkyWalker’s EpochOfEntry is “now”
equJ2000	2	Reported when SkyWalker’s “EpochOfEntry” is “2000” (i.e. J2000 equator/equinox, ICRS reference frame).
equJ2050	3	Reported when SkyWalker’s “EpochOfEntry” is “2050” (i.e. J2050 equator/equinox, ICRS reference frame).
equOther	0	Reported when SkyWalker’s “EpochOfEntry” is none of the above three values (e.g. “2010”).

Support when written

The ASCOM standard specifies that EquatorialSystem is read-only. Maestro allows a client to write EquatorialSystem. When written to one of the above values (equOther is not supported), SkyWalker’s EpochOfEntry is appropriately set.

Note: SkyWalker always reports **output** coordinates (available through the RightAscension and Declination properties) in local topocentric form (i.e. in the Epoch of “now”). EquatorialSystem only applies to the **input** format SkyWalker uses.

FocalLength: The telescope's focal length, meters.

Support when read

Not currently supported.

Support when written

Not a writeable property.

GuideRateDeclination: The current Declination movement rate offset for telescope guiding (degrees/sec). This is for use with the PulseGuide() method. With Maestro, the motion initiated by the PulseGuide() method moves at SkyWalker’s currently selected View Velocity.

Maestro support for ASCOM

Support when read

Not currently supported.

Support when written

Not currently supported.

GuideRateRightAscension: The current Right Ascension movement rate offset for telescope guiding (degrees/sec). This is for use with the PulseGuide() method. With Maestro, the motion initiated by the PulseGuide() method moves at SkyWalker's currently selected View Velocity.

Support when read

Not currently supported.

Support when written

Not currently supported.

InterfaceVersion: The version of this interface. Will return 2 for this version.

Support when read

Returns 2. Used to distinguish this driver from a v1.0 driver.

Support when written

Not a writeable property.

IsPulseGuiding: True if a PulseGuide() command is in progress, False otherwise.

Support when read

True if a PulseGuide() command is in progress, False otherwise.

Support when written

Not a writeable property.

Name: The short name of the telescope, for display purposes.

Support when read

Maestro reports "SkyWalker, via Maestro, controlling unknown telescope."

Support when written

Not a writeable property.

RightAscension: The right ascension (hours) of the telescope's current local equatorial coordinates.

Support when read

Fully supported. If SkyWalker is not Celestial Aligned then 0.0 is return.

Support when written

Not a writeable property.

Note: Right Ascension is provided in the Apparent Right Ascension form. This is in the form of "local topocentric coordinate system" in ASCOM terms (see more information under the EquatorialSystem property). SkyWalker does not

Maestro support for ASCOM

“backwards” correct Apparent celestial coordinates to standard reference frame (i.e. so called “catalog” coordinates in SkyWalker).

RightAscensionRate: The right ascension tracking rate offset from sidereal (seconds per sidereal second).

Support when read

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

RA seconds per hour = $R \div 15$ arcseconds per RA second

RA seconds per second = RA seconds per hour \div 3600 seconds per hour

RA seconds per sidereal second = RA seconds per second \times 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 RA seconds per sidereal second, is converted to SkyWalker’s required units of arcseconds per standard hour (per the reverse of the above sequence) and SkyWalker’s Custom TrackRate in RA is set accordingly.
2. If the property value is set to a non-zero value SkyWalker’s TrackRate is set to Custom.

RightAscensionRate is bounded to +/-18.46 seconds per sidereal second

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

Note: SkyWalker’s Custom TrackRate is the same as Sidereal prior to SkyWalker being aligned.

SideOfPier: Indicates which side of the pier a German equatorial mount is currently on.

Support when read

Fully supported. Behavior depends on **AlignmentMode**:

- If AlignmentMode = algGermanPolar: Returns pierEast (0) or pierWest (1).
- If AlignmentMode != algGermanPolar: error with the code 0x80040410 is raised with the message "Not available unless Meridian Avoidance is Full(GEM).".

Note: ASCOM nomenclature uses side of pier that the telescope is on. SkyWalker 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 currently supported.

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

Support when read

Fully supported.

Support when written

Maestro support for ASCOM

Not a writeable property.

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

Support when read

Not currently supported.

Note: SkyWalker uses Atmospheric Pressure and Temperature for refraction correction: not elevation above sea level.

Support when written

Not a writeable property.

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

Support when read

Fully supported.

Support when written

Fully supported.

Note: writing this property will void alignment

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

Support when read

Fully supported.

Support when written

Fully supported.

Note: writing this property will void alignment

Slewing: True if telescope is not currently moving in response to one of the Slew methods or the MoveAxis() method, False at all other times.

Support when read

True if SkyWalker's MoveMode is GoTo and False otherwise.

Support when written

Not a writeable 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.

Note: It is possible, particularly on slow PCs, that the coordinates read from Maestr/SkyWalker will not match the target coordinates of the slew for even a few seconds after Slewing is de-asserted. This is due to latencies in Window's processing and communications. Use SlewSettleTime if telescope coordinates must match target coordinates after Slewing de-assertion.

Note: Maestro uses the algorithm that Slewing is set if SkyWalker's MoveMode does not contain the work "Track" (e.g. "Track/Sidereal").

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

Support when read

Maestro support for ASCOM

Fully supported. The non-async slewing methods (FindHome(), Park(), SlewToAltAz(), SlewToCoordinates() and SlewToTarget()) will not return until the specified number of seconds after the operation within SkyWalker has completed.

Support when written

Fully supported.

Note: a non-zero SlewSettleTime should be used if the telescope coordinates must match the slew target coordinates at the completion of a slew operation or at the de-assertion of Slewing. A SlewSettleTime of up to 5 seconds may be necessary on very slow PCs. After sufficient SlewSettleTime the telescope and target coordinates will match.

TargetDeclination: The declination (degrees, positive North) for the target of a slew or sync operation.

Support when read

Fully supported.

Support when written

Fully supported

Note: target equatorial coordinates are assumed valid for SkyWalker's EpochOfEntry. See the EquatorialSystem property for more information.

TargetRightAscension: The right ascension (hours) for the target of a slew or sync operation.

Support when read

Fully supported.

Support when written

Fully supported

Note: target equatorial coordinates are assumed valid for SkyWalker's EpochOfEntry. See the EquatorialSystem property for more information.

Tracking: The state of the telescope's sidereal tracking drive.

Support when read

ASCOM value	SkyWalker 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 Lunar, Solar, etc.)

Support when written

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

Maestro support for ASCOM

Note: whenever this property is written to True SkyWalker's TrackRate will be set to Custom **if** either the DeclinationRate or RightAscensionRate properties are non-zero, otherwise SkyWalker's TrackRate will be reverted to the value saved when Tracking was set to False (as described above).

TrackingRate: The current tracking rate of the telescope's sidereal drive

Support when read

Provides the present TrackRate within SkyWalker:

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

Note: SkyWalker does not have a separate King (ASCOM driveKing) tracking rate. Instead, if refraction correction is enabled, SkyWalker's 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 SkyWalker is using its Custom TrackRate, then the RightAscension rate and DeclinationRate should be polled... if either are non-zero then SkyWalker is using its Custom TrackRate.

Note: if SkyWalker's TrackRate is Drift then an error with the code 0x80040410 will be raised with the message "SkyWalker's TrackingRate is Drift (no ASCOM code exists for this)". ASCOM does not use SkyWalker's 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 SkyWalker's TrackRate will change accordingly.

TrackingRates: Returns a collection of supported tracking rate values that describe the permissible values of the TrackingRate property.

Support when read

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

Support when written

Not a writeable property.

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

Support when read

Returns the UTC time/date read from SkyWalker.

Support when written

When this property is written, SkyWalker's internal RTCC (Read Time Clock/Calendar) is set.

Note: SkyWalker's RTCC is set to local standard time/date. To set SkyWalker's RTCC, Maestro uses the GMT Offset value to calculate local standard time/date. Therefore, before writing this property assure that SkyWalker's GMT Offset is correctly set.

Note: writing this property will void alignment.

Note: for a more accurate means of setting SkyWalker's time from PC time refer to the section "Using SkyWalker/Maestro Remotely with ASCOM" under the topic of "High accuracy PC -to- SkyWalker time sync".

Implementation of ASCOM Telescope Interface Methods

AbortSlew(): Stops a slew (i.e. GoTo) in progress.

Fully compliant.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

AxisRates(Axis): Determine the rates at which the telescope may be moved about the specified axis by the MoveAxis() method.

Not supported. Returns an empty collection and no error is raised when called (both per spec).

CanMoveAxis(Axis): True if the telescope can be controlled about the specified axis via the MoveAxis() method.

Not supported. Returns False and no error is raised when called (both per spec).

CommandBlind(Command [, Raw]): Send a string command directly to the telescope without expecting response data.

Not supported.

CommandBool(Command [, Raw]): Send a string command to the telescope, returning a true/false response.

Maestro uses the CommandBool() method to provide settings useful for remote operation, including:

- Settings that disable interactive dialog at the telescope
- Settings that specify which messages Maestro will raise as trappable errors
- Synchronizing SkyWalker's time/date to Maestro/PC time/date

See the sections "Using SkyWalker/Maestro Remotely with ASCOM" and "SkyWalker Asynchronous Message Reporting to ASCOM Client Applications" for complete details on Maestro's use of the CommandBool() method.

Note: the Raw parameter is ignored.

CommandString(Command [, Raw]): Send a string command to the telescope, returning a response string.

Maestro uses the CommandString() method to:

- Allow the Client to send ATCL string commands directly to SkyWalker. SkyWalker's response to the ATCL command is provided in the response string.

Note: SkyWalker's primary interface protocol is called the Astrometric Telescope Control Language (ATCL). ATCL is typically communicated over serial link (e.g. RS232 or RS422/USB/Ethernet with suitable converters) to SkyWalker's Client interface port (labeled "Com"). ATCL provides a very rich syntax for telescope control. Maestro controls SkyWalker through ATCL and the CommandString() method allows ASCOM Clients direct access to the ATCL interface to SkyWalker.

To pass an ATCL command directly to SkyWalker, call the CommandString() method with a string parameter that is an ATCL command. The response string of the CommandString()

function is the response from SkyWalker. For example, to set SkyWalker's ViewVel4 to 0.8deg/sec you could use the following VBA CommandString() code:

```
Set Scope = CreateObject( "Maestro.Telescope" )  
Response = Scope.CommandString( "KSv40.8deg/sec" )
```

More information on ATCL is provided at www.astrometric.com/support/resources/ATCL.html.

Note: the Raw parameter is ignored however the behavior of this command is that the Command string is always sent to SkyWalker, 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 this 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. For these ATCL commands, the CommandString() method will return a one-character string with the ATCL_ACK or ATCL_NACK character.
-
- Query the current message level. See the section "SkyWalker Asynchronous Message Reporting to ASCOM Client Applications" for complete details

DestinationSideOfPier(RightAscension, Declination): Predict side of pier for German equatorial mounts.

Not supported.

FindHome(): Locates the telescope's "home" position.

Fully compliant. This method does not return until the home position has been found (i.e. it is "synchronous").

Note: this method requires that an optional Home Index Sensor (and Automation Module support) is installed.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

MoveAxis(Axis, Rate): Move the telescope in one axis at the given rate.

Not supported.

Park(): Move the telescope to its park position and stop tracking motion. Returns when the GoToPark is complete.

Fully compliant.

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 SkyWalker then SkyWalker will generate the error "Not available until parked position set.". This error will be propagated through Maestro to the ASCOM Client with a code of 0x80040411 and will not be raised to the client unless the Message Level (see last section of this document) is set to L1 or L2.

PulseGuide(Direction, Duration): Moves the scope in the given direction for the given interval or time at the rate given by the corresponding guide rate property.

Partially compliant. Causes motion at SkyWalker's current View Velocity but **not** at the rate given by the corresponding guide rate property.

This method returns immediately so that PulseGuide'ing in both directions can be initiated.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

SetPark(): Sets the telescope's park position to be its current position.

Fully compliant.

SetupDialog(): Displays a setup dialog, allowing the user to set telescope-specific values such as baud rate, geodetic position, etc.

Not supported. Maestro presents a dialog box stating "Please use Maestro to make settings for connection-to and use-of SkyWalker." when this method is called.

SlewToAltAz(Azimuth, Altitude): Move the telescope to the given coordinates, return when slew is complete.

Fully supported.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

SlewToAltAzAsync(Azimuth, Altitude): Move the telescope to the given coordinates, return immediately after starting the slew.

Fully supported.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

SlewToCoordinates(RightAscension, Declination): Move the telescope to the given coordinates, return when slew is complete.

Fully supported.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

SlewToCoordinatesAsync(RightAscension, Declination): Move the telescope to the given coordinates, return immediately after starting the slew.

Fully supported.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

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

Fully supported.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True.".

Maestro support for ASCOM

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

Fully supported. If the target coordinates have not be set then a trappable error with the code 0x80040405 and the message: "TargetRightAscension <TargetDeclination> ASCOM property has not yet been set." is raised.

Note: if the AtPark property is True, calling this method will raise an error with the code 0x80040410 and the message "Action not allowed when AtPark is True."

SyncToAltAz(Azimuth, Altitude): Matches the scope's local horizontal coordinates to the given local horizontal coordinates.

Fully supported.

Note: this is not allowed if AtPark is True or Tracking is True (per ASCOM spec).

Note: If a "Synch" command is received when SkyWalker 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.

SyncToCoordinates(RightAscension, Declination): Matches the scope's equatorial coordinates to the given equatorial coordinates.

Fully supported.

Note: this is not allowed if AtPark is True or Tracking is False (per ASCOM spec).

Note: If a "Synch" command is received when SkyWalker 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.

SyncToTarget(): Matches the telescope's current coordinates to Telescope.TargetRightAscension and Telescope.TargetDeclination.

Fully supported. If the target coordinates have not be set then a trappable error with the code 0x80040405 and the message: "TargetRightAscension <TargetDeclination> ASCOM property has not yet been set." is raised.

Note: this is not allowed if AtPark is True or Tracking is False (per ASCOM spec).

Note: If a "Synch" command is received when SkyWalker 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.

Unpark(): Takes the telescope out of the Parked state. Start tracking from the parked position, valid only after Park().

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

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

Note: When the telescope is AtPark, there is no movement allowed from an ASCOM Client (required per the spec) however movement from handpaddle or Maestro is still allowed (since it does not impact the Align-from-last accuracy) and can result in the telescope no longer being AtPark even if no movement was requested from an ASCOM client .

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

Using SkyWalker/Maestro Remotely with ASCOM

To effectively use a telescope control system remotely there must be

1. Means to turn-off interactive “dialog” at the control system console (i.e. at the telescope).
2. Means to receive messages back from the telescope control system which (can) report on operational state of the instrument (i.e. so that a client need not poll every conceivable operational status all the time to check for fault, error and status events).
3. Means to synchronize SkyWalker’s time to Maestro/PC time with minimal error due to processing latency.

The second requirement is discussed in the next section, the other two are discussed here...

Messaging back to the client

For remote operation, Maestro’s “interactive dialog” must be disabled since there is no user at the instrument to carryout the dialog. To disable interactive dialog call the **CommandBool()** method with “ID0” as the parameter. To re-enable interactive dialog call the **CommandBool()** method with “ID1” as a parameter. The return value in both cases is True. If an unknown string is passed to **CommandBool()** then the return value is False.

There are a few SkyWalker features that are disabled when interactive dialog is disabled:

- AutoCalibration: because AutoCalibration requires interactive dialog with the user where the user acknowledges that the calibration star is centered prior to completion of the GoTo.
- Calibration sanity checking: this feature, which checks that calibrations do not involve >10degrees of “disparity” in either RA or Dec from current position, is disabled because it queries the user.

High accuracy PC –to- SkyWalker time sync

To synchronize SkyWalker’s time to Maestro/PC time call the **CommandBool()** method with “TSync” as the parameter. This signals Maestro to update SkyWalker time and date to the PC’s time/date in a low latency manner. Note: this will also void alignment.

Writing the **UTCDate** property will also set SkyWalker time/date however there can be a few seconds of latency since the time/date is sourced from the ASCOM Client, passed to Maestro and then to SkyWalker.

SkyWalker’s Asynchronous Message Reporting to ASCOM Client Applications

The Maestro ASCOM Telescope Interface driver provides a means for Maestro to “raise” messages, generated by SkyWalker, to the ASCOM client as trappable errors. SkyWalker messages include:

- Status (e.g. “Passed through Celestial Pole”)
- Warnings (e.g. “Approaching Meridian Limit”)
- Alerts (e.g. “Dec/Alt HardLimit condition entered”)
- Fatal errors. These include “Internal Errors” (raised from internal code error checking) and client syntax errors (where an ATCL command to SkyWalker is illegal).

Maestro support for ASCOM

To enable Maestro to handle messages as trappable errors the message “level” that Maestro should raise needs to be specified. Use the **CommandBool()** method to specify to Maestro what message level to raise. The command string to use with CommandBool(), associated message levels, and the definition of the message levels are contained in the table below.

Command string	Message Level	Definition of Message Level	CommandBool() return value
“L1”	Level 1	Causes Maestro to raise ALL messages as trappable errors.	True
“L2”	Level 2	Causes Maestro to raise only warnings, alerts and fatal errors.	True
“L3”	Level 3	Causes Maestro to raise only alerts and fatal errors.	True
“L4”	Level 4	Causes Maestro to raise only fatal errors.	True

At start-up, Maestro’s message level is set to L4.

There are two methods to query Maestro’s current message level:

1. Call the **CommandBool()** method as follows: CommandBool(“Ln?”) where ‘n’ is the message level for which a query is being made. The return string will be True if the message level is “Ln” and False otherwise. For example, if the message level is presently 3 then CommandBool(“L1?”) will return False and CommandBool(“L3?”) will return True.
2. Call the **CommandString()** method as follows: CommandString(“L?”). The return string will be “L1” through “L4” indicating the current message level.

The trappable error code that Maestro raises for messages is 0x80040411. The message associated with this error is simply the text of the message directly from SkyWalker.

Maestro raises SkyWalker’s messages by “throwing” a (trappable) run-time error back to the client application (with an error code of 0x80040411). Therefore, if a client application enables Maestro to raise errors it **must** have an active error handler to trap/report the error otherwise the error will be raised to the operating system and terminate the client program and/or script execution.

The Maestro ASCOM Telescope Interface driver will only raise a trappable error when one of the ASCOM telescope properties or methods (for which Maestro provides support) is accessed. This prevents the ASCOM client from being potentially flooded with trappable errors. Therefore, to assure that messages are not missed the ASCOM telescope properties and methods should be polled and/or called on a regular basis (say once every second). An active ASCOM client is likely to do this anyhow to receive, for example, up-to-date celestial coordinates.

Note: any message that Maestro receives from SkyWalker during Maestro’s processing of the property or method is raised to the client however any message arriving from SkyWalker **after** Maestro has returned from the property/method (for example: a SlewTo hits a limit after it has started) will not be raised to the client until the next property/method access.

More Resources

Complete details on ASCOM standards can be found at <http://ascom-standards.org/>. This includes information on developing client software and/or scripts.

Maestro support for ASCOM

Visit our www.astrometric.com/support/resources/ASCOM.html page for the most up-to-date version of this document and a complete set of Visual Basic for Applications scripts that exercise all the ASCOM Telescope Interface Standard properties and methods supported by Maestro.