

**EQASCOM: Supported ASCOM Properties and Methods.**

<b>ASCOM Properties</b>	<b>Required/ Optional</b>	<b>EQASCOM</b>	<b>Read</b>	<b>Write</b>
AlignmentMode	O	✓	✓	N/A
Altitude	O	✓	✓	N/A
ApertureArea	O	X		
AperertureDiameter	O	X		
AtHome	R	✓	✓	N/A
AtPark	R	✓	✓	N/A
Azimuth	O	✓	✓	N/A
CanFindHome	R	✓	✓	N/A
CanPark	R	✓	✓	N/A
CanPulseGuide	R	✓	✓	N/A
CanSetDeclinationRate	R	✓	✓	N/A
CanSetGuideRates	R	✓	✓	N/A
CanSetPark	R	✓	✓	N/A
CanSetPierSide	R	✓	✓	N/A
CanSetRightAscensionRate	R	✓	✓	N/A
CanSetTracking	R	✓	✓	N/A
CanSlew	R	✓	✓	N/A
CanSlewAltAz	R	✓	✓	N/A
CaqnSlewAltAxAsync	R	✓	✓	N/A
N/A CanSlewAsync	R	✓	✓	N/A
CanSync	R	✓	✓	N/A
CanSyncAltAz	R	✓	✓	N/A
CanUnpark	R	✓	✓	N/A
Connected	R	✓	✓	✓
Declination	R	✓	✓	N/A
DeclinationRate	O	✓	✓	✓
Description	R	✓	✓	N/A
DoesRefraction	O	X		
DriverInfo	R	✓	✓	N/A
DriverVersion	R	✓	✓	N/A
EquatorialSystem	R		✓	N/A
FocalLength	O	X		
GuideRateDeclination	O	✓	✓	✓
GuideRateRightAscension	O	✓	✓	✓
InterfaceVersion	R	✓	✓	N/A
IsPulseGuiding	O	✓	✓	N/A
Name	R	✓	✓	N/A
RightAscension	R	✓	✓	N/A
RightAscensionRate	O	✓	✓	✓
SideOfPier	O	✓	✓	X
SiderealTime	R	✓	✓	N/A
SiteElevation	O	✓	✓	X
SiteLongitude	O	✓	✓	X
Slewing	O	✓	✓	N/A

SlewSettleTime	O	✓	✓	✓
TargetDeclination	O	✓	✓	✓
TargetRightAscension	O	✓	✓	✓
Tracking	R	✓	✓	✓
TrackingRate	O	✓	✓	X
TrackingRates	O	✓	✓	X
UTCDate	R	✓	✓	X

ASCOM Methods	Required /Optional	EQASCOM		
AbortSlew	O	✓		
AxisRates	R	✓		
CanMoveAxis	R	✓		
CommandBlind	O	X		
CommandBool	O	X		
CommandString	O	✓		
DestinationSiderOfPier	O	✓		
FindHome	O	✓		
MoveAxis	O	✓		
Park*	O	✓		
PulseGuide	O	✓		
SetPark	O	✓		
SetupDialog	R	✓		
SlewToAltAz	O	X		
SlewToAltAzAsync	O	X		
SlewToCoordinates	O	✓		
SlewToCoordinatesAsync	O	✓		
SlewToTarget	O	✓		
SlewToTargetAsync	O	✓		
SyncToAltAz	O	X		
SyncToCoordinates	O	✓		
SyncToTarget	O	✓		
Unpark	O	✓		

\* Please note that by default EQASOM implements the park method as an ASYNCHRONOUS function. Clients can simply issue a park command and then poll the AtPark property to determine when a park operation has completed. Although this works well in most case it would seem that ASCOM intended for Park to be an SYNCHRONOUS method (although the ASCOM specification is woefully unclear on this). Synchronous methods only return control to the clients once the method has completed and as parking can take quite some time this actually seems a poor design decision when applied to mount control software. However, we are aware of some client applications that rely synchronous parking and so we provide an option in the EQASCOM setup to allow this type of park to be performed.

## EQASCOM CommandString Interface

Whilst ASCOM exposes many driver and mount specific properties and functions it is by no means a comprehensive standard (or indeed a particularly consistent one) and there is a general reluctance by those guiding ASCOM to expand or update the standard.

The CommandString interface provides a method by which a driver can expose additional proprietary properties and functions. Developers of ASCOM client applications (or indeed those using the standard windows Component Object Model) are welcome to access these additional properties and functions but must be aware that these are specific only to EQASCOM and are unlikely to be found on other ASCOM drivers.

Function	Command String	EQASCOM Responds
Disable PEC	:PECENA,0#	
Enable PEC	:PECENA,1#	
Get PEC State	:PECSTA#	0# (PEC Disabled) 1# (PEC Enabled)
Get Worm Tooth Count	:PECWTC#	worm_tooth_count#
Get Worm position	:PECIDX#	worm_position#
Get PEC Info	:PECINFO#	row_count,max_position #
Set Table Row	:PECSET,row_index,worm_position,pe #	1# (success) 0# (failure)
Get Table Row	:PECSET,row_index#	1,worm_position,pe# 0# (failure)
Load PEC Table	:PECLOAD,full_file_name#	1# (success) 0# (failure)
Save PEC Table	:PECSAVE,full_file_name#	1# (success) 0# (failure)
Get PEC Gain	:PECGAIN	gain#
Set PEC Gain	:PECGAIN,gain#	1# (success) 0# (failure)
Get PEC Phase	:PECPHASE	phase#
Set PEC Phase	:PECPHASE,phase#	1# (success) 0# (failure)
Get Mount Version	:MOUNTVER#	MountVersionString
Get EQASCOM Version	:DRIVERVER#	EQASCOM Version
Get eqcontrl.dll version	:DLLVER#	Dll version
Park	:PARK,parkmode#	1# (mount parked or parking) 0# mount unparked

Unpark	:UNPARK,unparkmode#	1# (mount unparked or unparking) 0# (mount parked)
Get RA encoder	:RA_ENC#	Encoder position
Get DEC encoder	:DEC_ENC#	Encoder position
Get ST4 RA Guide Rate	:ST4_RARATE#	ST4GuideRate#
Set ST4 RA Guide Rate	:ST4_RARATE,ST4GuideRate#	1# (success) 0# (failure)
Get ST4 DEC Guide Rate	:ST4_DECRATE#	ST4GuideRate#
Set ST4 DEC Guide Rate	:ST4_DECRATE,ST4GuideRate#	1# (success) 0# (failure)
Get PulseGuide RA Guide Rate	:PG_RARATE#	PGGuideRate#
Set PulseGuide RA Guide Rate	:PG_RARATE,PGGuideRate#	1# (success) 0# (failure)
Get PulseGuide DEC Guide Rate	:PG_DECRATE#	PGGuideRate#
Set PulseGuide DEC Guide Rate	:PG_DECRATE,PGGuideRate#	1# (success) 0# (failure)
Get Alignment mode	:ALIGN_MODE#	1# (append) 0# (dialog)
Set Alignment mode	:ALIGN_MODE,0# :ALIGN_MODE,1#	1# (success)
Clear sync	:ALIGN_CLEAR_SYNC#	1# (success) 0# (failure)
Clear points	:ALIGN_CLEAR_POINTS#	1# (success) 0# (failure)
Get sync limit status	:ALIGN_SYNC_LIMIT#	1# (active) 0# (inactive)
Set sync limit status	:ALIGN_SYNC_LIMIT,0# :ALIGN_SYNC_LIMIT,1#	1# (success) 0# (failure)
Get Flipped Goto status	:FLIP_GOTO#	1# (active) 0# (inactive)
Set Flipped Goto status	:FLIP_GOTO,0# :FLIP_GOTO,1#	1# (success) 0# (failure)

All commands from the client begin with “:”  
All commands end with “#”

Parameters are comma separated and passed as ASCII encoded integers. Avoid using any locale grouping settings (i.e. send 50132 rather than 50,132 ).

worm_tooth_count	Number of teeth on RA axis gear.
row_count	Number or rows in the PEC table. Equates to the wormperiod.
max_position	Maximum worm position = microsteps per worm revolution –1
worm_position	Current worm position ( 0 to max_position)
pe	Periodic error * 1000
row_index	PEC table index ( 0 to row_count –1)
full_file_name	Full file path.
MountVersionString	XX.YY.ZZ  XX = major version (hex encoded) YY = Minor Version (hex encoded) ZZ = subversion (hex encoded)
parkmode	Numeric value as follows: 0 = park using current EQASCOM park operation 1 = Park to Home position 2 = Park to current Position 3 = Park to user position 1 4 = Park to user position 2 5 = Park to user position 3 6 = Park to user position 4 7 = Park to user position 5  note that not all user positions may be defined – check the response (1# or 0#) to determine success.
unparkmode	Numeric value as follows: 0 = unpark using current EQASCOM unpark operation 1 = unpark to current 2 = unpark to user position 1 3 = unpark to user position 2 4 = unpark to user position 3 5 = unpark to user position 4 6 = unpark to user position 5  note that EQASCOM provides up to 5 user defined park positions not all user positions may be defined – check the response (1# or 0#) to determine success.
ST4GuideRate	Either 0.25 0.5 0.75 1.00
PGGuideRate	Either 0.1

	0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

Writes to the EQASCOM PEC table using the PECSET command are buffered into a temporary table. When the last row is written the table is automatically saved to disk as %APPDATA%\EQMOD\pec.txt and then loaded as the current PEC definition. EQASCOM will then use this file location for future automatic loads until such time as the user manually specifies an alternate filepath.

Reads of the EQASCOM PEC table using the PECGET command read the specified row of the currently loaded PEC file.

Steps per worm revolution, Worm period and Worm tooth count are all determined from parameters read from the mount itself. If the mount (or simulator) is not currently connected then reading these parameters will result in a value of -1.