

tcs.server

Interface ScopeInterface

```
public interface ScopeInterface
```

Interface to telescope control.

Field Summary

static java.lang.String	HandlerName
-------------------------	-----------------------------

Method Summary

boolean	controlPointingModel (java.lang.String cmd) Turn pointing correction on/off or (re)load Pointing Model coefficients from conf-file
java.lang.String	getPointingModel () Get Pointing Model current state
java.lang.String	getStatusString (java.lang.String request) Obtaining the list of Telescope parameters
boolean	isSimulator ()
boolean	setCorr (java.lang.String cor_params) Modify telescope runtime corrections
boolean	setMaxVel (double mxvHA, double mxvDecl) Set Max.Speed restrictions for telescope axes
boolean	setMovingVel (double V_ha, double V_decl) set velocities for manual telescope moving (stop current observation if it in progress now)
boolean	setObjectVel (double V_ha, double V_decl) set tracking velocities for moving objects (like comets, asteroids, satellites)
boolean	setShiftingVel (double V_ha, double V_decl) set shifting velocities to correct object coordinates
boolean	setupObject (java.lang.String obj_params) Set observing object parameters
boolean	slewToObject () Start telescope slewing to new observing object

boolean	slewToPosition() Start telescope slewing to fixed position
boolean	telescopeStop() Stop telescope
boolean	telescopeStopTrack() Stop telescope moving but start HA-tracking drive
boolean	toHardware() Switch telescope control to "Hardware" mode (real SEW-drives&Encoders usage)
boolean	toSimulator() Switch telescope control to "Software Simulation" mode
boolean	touch() Inform server that client is Ok

Field Detail

HandlerName

static final java.lang.String **HandlerName**

See Also:

[Constant Field Values](#)

Method Detail

touch

boolean **touch()**

Inform server that client is Ok

Returns:

true if telescope server is alive

toSimulator

boolean **toSimulator()**

Switch telescope control to "Software Simulation" mode

Returns:

false if already this mode

toHardware

boolean **toHardware**()

Switch telescope control to "Hardware" mode (real SEW-drives&Encoders usage)

Returns:

false if already this mode

isSimulator

boolean **isSimulator**()

Returns:

true if telescope control is in "Simulation" mode

setupObject

boolean **setupObject**(java.lang.String obj_params)

Set observing object parameters

Parameters:

obj_params - list of object parameters values in form: "... Name=Value ..."

required:

RA=hh:mm:ss.ss Decl=+dd:mm:ss.s

additional possible parameters:

Epoch=dddd.d [2000,0] epoch (and equinox) of coordinates

PMRA=d.d [0.0] proper motion on R.A. (mas/year *cos(Decl))

PMDecl=d.d [0.0] proper motion on Declination (mas/year)

Parallax=d.d [0.0] parallax (arcsec)

RadVel=d.d [0.0] radial velocity (km/sec)

WaveLen=0.ddd [0.550] observation effective wavelength (micron)

special cases:

AppRA=hh:mm:ss.ss AppDecl=+dd:mm:ss.s - for pointing to "apparent place"

ObsRA=hh:mm:ss.ss ObsDecl=+dd:mm:ss.s - for pointing to "observed place"

HA=hh:mm:ss.ss Decl=+dd:mm:ss.s - for pointing to H.A./Decl position

Az=ddd:mm:ss.s ZD=+dd:mm:ss.s - for pointing to Azimuth/Zen.Dist position

Returns:

true - if object coordinates in observable area, else - false

slewToObject

boolean **slewToObject**()

Start telescope slewing to new observing object

Returns:

true

slewToPosition

boolean **slewToPosition**()

Start telescope slewing to fixed position

Returns:

true

telescopeStop

boolean **telescopeStop**()

Stop telescope

Returns:

true

telescopeStopTrack

boolean **telescopeStopTrack**()

Stop telescope moving but start HA-tracking drive

Returns:

true

setObjectVel

boolean **setObjectVel**(double V_ha,
double V_decl)

set tracking velocities for moving objects (like comets, asteroids, satellites)

Parameters:

V_{ha} - object tracking velocity (in addition to sky=1.0) for HA-axis (sec/sec)

V_{decl} - object tracking velocity for Decl-axis (arcsec/sec)

Returns:

true

setShiftingVel

boolean **setShiftingVel**(double V_{ha} ,
double V_{decl})

set shifting velocities to correct object coordinates

Parameters:

V_{ha} - object shifting velocity for HA-axis (sec/sec)

V_{decl} - object shifting velocity for Decl-axis (arcsec/sec)

setMaxVel

boolean **setMaxVel**(double mxvHA,
double mxvDecl)

Set Max.Speed restrictions for telescope axes

Parameters:

mxvHA - H.A maximum velocity value (sec/sec)

mxvDecl - Decl. maximum velocity value (arcsec/sec)

Returns:

true

setMovingVel

boolean **setMovingVel**(double V_{ha} ,
double V_{decl})

set velocities for manual telescope moving
(stop current observation if it in progress now)

Parameters:

V_{ha} - Hour-axis velocity (sec/sec)

V_{decl} - Decl-axis velocity (arcsec/sec)

Returns:

true

setCorrboolean **setCorr**(java.lang.String cor_params)

Modify telescope runtime corrections

Parameters:

cor_params - list of correction values in form: "... Name=Value ..."

Examples:

ObjHAadd=0.1 ObjDecladd=1.5

TelHAvel=0.1 TelDeclvel=0.5

HdwHASET=1.1 HdwDeclset=20.5

Names syntax:

Obj... - Sky Object coordinates correction (not in "Stop" mode)

Tel... - Telescope position correction (only in "Tracking" mode)

Hdw... - Acquisition Hardware Opt.axis.position correction

...HA... - HourAngle correction (sec)

...RA... - R.A. (opposite to HA) correction (sec)

...Decl... - Declination correction (arcsec)

...set - set new correction value

...add - add correction to current value

...vel - set correction velocity (sec/sec or arcsec/sec)

Returns:

false - if can't execute corrections

getStatusStringjava.lang.String **getStatusString**(java.lang.String request)

Obtaining the list of Telescope parameters

Parameters:

request - parameters names list

Returns:

list of values requested (".... Name=Value")

Possible Names of requested parameters:

Status - [Off|DrvOff|EncOff|Stop|Moving|Slewing|Tracking]

Tel... - Telescope Object parameters

Obj... - Observed Object parameters

Inp... - Input Object parameters

...Mean... - mean RA/Decl/ST

...App... - apparent RA/Decl/ST/MJD

...Obs... - observed RA/HA/Decl/Az/ZD/MJD
...Enc... - encoders HA/Decl
...Epoch - epoch & equinox of Object mean coordinates
...MJD - modified Julian date of Object App/Obs coordinates
...RA... - Object Mean/App/Obs R.A.
...Decl... - Object Mean/App/Obs/Enc Declination
...HA... - Object Obs/Enc HourAngle
...Az... - Object Obs azimuth
...ZD... - Object Obs zenith distance
...Expos - Object current exposure (remaining) time
...Rever - Object reversal flag ("Yes"/"No")
...PMHA - Pointing Model correction value for HourAngle
...PMDecl - Pointing Model correction value for Declination
Vel... - Telescope HA/Decl drives current velocity
MaxVel... - Telescope HA/Decl Max.Speed restrictions
Deviat... - HA/Decl angular deviation i.e. object and telescope mismatch (discrepancy)
...str - sexagesimal format of time/angle values, (float - otherwise)

...corr - HA/Decl runtime corrections:
Tel...corr - for "index" (telescope encoders) correction
Hdw...corr - for "collimation" (Acquisition Hardware Opt.axis) correction

controlPointingModel

boolean **controlPointingModel**(java.lang.String cmd)

Turn pointing correction on/off or (re)load Pointing Model coefficients from conf-file

Parameters:

cmd - "On" or "Off" or "Load"

Returns:

true if recognize cmd as correct command

getPointingModel

java.lang.String **getPointingModel**()

Get Pointing Model current state

Returns:

"State=.. IH=.. TelPMDecl=.."

