Robert C. Byrd Green Bank Telescope
NRAO Green Bank

Toney Minter and Dana Balser

17th September 2003

GBT SOFTWARE PROJECT NOTE 21.2

Commonly Used GO Keywords

HTML version Available[1]

Contents

[1] http://www.gb.nrao.edu/GBT/MC/doc/dataproc/gbtGOKeywords/gbtGOKeywordsManual/gbtGOKeywordsManual.html

Abstract

This document describes the GO Keywords that will be commonly used in GO Tables by observers. GO Keywords are used within GO Tables to setup the GBT for observations. The keywords listed in this document do not comprise the total list of GO keywords. We also present the possible values for the keywords.

N.B. This document will be continuously under construction while GO is being developed. Thus the reader should frequently check for new versions of this document.

History

13th May 2002 Add mapping information to included tex file and the ability to ignore that information in this file.

17th September 2003 Bring document up to date. List all reasonable keywords available from GO.

1 Introduction

Every setup and built-in procedure parameter in the GBT Observer's interface (GO) has a predefined keyword. When a value is assigned to a keyword in a GO table the corresponding parameter(s) are set in hardware or the value is retained for use when a procedure is called. Most keywords have unique names in the system, but some, like bandwidth, are repeated in different devices (spectral processor, spectrometer, etc.). Hence, every keyword is assigned to a group which is designated by a keyword prefix. For example, sp.bandwidth is the bandwidth of the spectral processor. If a keyword applies to only one device, the prefix may be omitted, but the safest thing is to use it. The prefixes assigned so far are listed below.

Keywords may be written in their shortest, unambiguous form, e.g. sp.band, but for table readability it is usually a good idea to use the full name or assign a good alias. The shortened form is better left to interactive assignment on the command line of the command line window where conflicts are caught immediately and help is available on the possible choices.

2 Keyword Prefixes

Keywords are associated with specific hardware devices or with the procedure set. A few keywords naturally have the same name on different devices so every keyword has a prefix to designate its device group, e.g. dcr.integrationtime. The prefixes assigned so far are shown in Table 1.

Keyword Prefix	Device
proc	Procedures
sc	Scan Coordinator
ant	Antenna
lo1	LO1
conv	Converter Rack
ifrack	IF Rack
algfilt	Analog Filter Rack
rx	Currently selected receiver
rxpf1	Prime Focus 1 Receiver
rxpf2	Prime Focus 2 Receiver
rx1to2	21 cm Receiver
rx2to3	11 cm Receiver
rx4to6	6 cm Receiver
rx8to10	3 cm Receiver
rx12to18	2 cm Receiver
rx18to26	1 cm Receiver
rx26to40	8 mm Receiver
rx40to52	6 mm Receiver
rx65to95	4 mm Receiver
rx95to115	3 mm Receiver
pfsup	Prime Focus Support Rack
bcpm	Berkley-CalTech Pulsar Machine
dcr	Digital Continuum Receiver
spm	Spectrometer
sp	Spectral Processor

3 GO Keyword Types

GO Keywords values can be of several types. These are:

floats Keywords of the float type can take on any numerical value. Sometimes the range of allowed values is limited. For example, the rest freqeuncy of the observations must be greater than zero.

integers Keywords of this type can only be integers. Sometimes the range of allowed values is limited.

strings Must be inclosed in a pair of double qoutes. For example,

sc.proj_id = "GBT-01A-075"

boolean Currently, consists of the two strings "T" and "F".

enumerated Some keywords are limited to a few possible values. These keywords are of the enumaration type. The possivle values consist of specific string values. For example, the 1 cm receiver beam switching parameter can be either

rx18to26.beam_ctrl = "computer"

or

rx18to26.beam_ctrl = "manual"

sexagecimal Presently, sexagesimal formats for time, R.A., and Dec. constants are HH:MM:SS.S and sDD:MM:SS.S. No whitespace is allowed in this type.

arrays Some keywords are actually arrays. For example, there are up to eight IF center frequencies that can be set in the spectral processor, one for each IF channel. As in glish, keyword arrays are subscripted with square brackets, and ranges may be specified. Some possibilities are

Set the second IF channel center frequency to 256.8 MHz.

sp.iffrequency[2] = 256.8

Set IF channels 1 through 4 to separate center frequencies.

sp.iffrequency[1:4] = [245.0, 255.0, 245.0, 255.0]

Set IF channels 1, 3, and 5 to separate center frequencies.

sp.iffrequency[1,3,5] = [245.0, 255.0, 245.0]

Set all IF center frequencies to 250.0 MHz.

sp.iffrequency = 250

If there is a mismatch between the number of indices in the keyword index array and the number of values to the right of the equal sign, you will get a warning, but the assignment will be executed anyway. Extra values will be ignored. If there are too few values, the last value will be assigned to all remaining keyword array members specified. More complex glish index syntax, such as [1,2,4:6], or wild cards are not recognized by the table parser. If the keyword is not an array, the index will be ignored.

A keyword array can be an array of any of the above types (floats, integers, strings, boolean or enumerated) except sexagecimal.

4 GO Keywords

4.1 Procedures

There are currently seventeen pre-defined procedures that can be used. To have any procedure run the observer just needs to enter the procedures name in a GO Table. All parameters that are used by that procedure must be set in the GO Table before the procedure is run. Below is a list of the pre-defined procedures and the values that they use.

4.1.1 List of available procedures

- track procedure
- cross procedure
- ralongmap procedure
- declatmap procedure
- pointmap procedure
- offonoff procedure
- fivepoint procedure
- circle procedure
- tipping procedure
- focusprime procedure
- peak procedure
- pointfocussubreflector procedure
- slewto pseudo-procedure
- offon procedure
- onoff procedure
- nod procedure
- focussubreflectorprocedure

4.1.2 track procedure

This procedure tracks a fixed point in the chosen coordinate system or moves the telescope from that point at a constant rate in one or both coordinates. The available keywords for the track procedure are shown in Table 1.

4.1.3 cross procedure

The Cross procedure sweeps through the specified position in the chosen coordinate system in the four cardinal directions. Its primary purpose to deterime pointing offsets. Each sweep will be a separate scan with the Scan Length automatically determined by the coordinate Lengths and Rates. The available keywords for the cross procedure are shown in Table 2.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.rarate	proc.harate	proc.longrate	proc.azrate	proc.udlongrate	proc.sslongrate
proc.decrate	proc.decrate	proc.latrate	proc.elevrate	proc.udlatrate	proc.sslatrate
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.scanduration					
proc.repeatnumber					
proc.realtimedisplay					
proc.data					

Table 1: Keywords available for the track procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.rarate	proc.harate	proc.longrate	proc.azrate	proc.udlongrate	proc.sslongrate
proc.decrate	proc.decrate	proc.latrate	proc.elevrate	proc.udlatrate	proc.sslatrate
proc.ralength	proc.halength	proc.longlength	proc.azlength	proc.udlonglength	proc.sslonglength
proc.declength	proc.declength	proc.latlength	proc.elevlength	proc.udlatlength	proc.sslatlength
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.startnumber					
proc.repeatnumber					
proc.realtimedisplay					
proc.data					

Table 2: Keywords available for the cross procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

4.1.4 ralongmap procedure

RA/Long. Map does a raster scan centered on the specified position in the chosen coordinate system. Scanning is in the R.A., longitude, or azimuth coordinate with the direction and starting corner determined by the signs of the Rate and Step size. The available keywords for the ralongmap procedure are shown in Table 3.

4.1.5 declatmap procedure

Dec/Lat. Map does a raster scan centered on the specified position in the chosen coordinate system. Scanning is in the Dec., latitude, or elevation coordinate with the direction and starting corner determined by the signs of the Rate and Step size. The available keywords for the declatmap procedure are shown in Table 4.

4.1.6 pointmap procedure

Point Map constructs a map by integrating at fixed positions on a grid. The stepping direction and starting corner are determined by the signs of the Step sizes. The extent of the map is determined by the Step sizes and the number of Points in each direction. The fastest stepping is in R.A., longitude, or azimuth. A reference off position and frequency of occurrence may be specified. The available keywords for the pointmap procedure are shown in Table 5.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.rarate	proc.harate	proc.longrate	proc.azrate	proc.udlongrate	proc.sslongrate
proc.ralength	proc.halength	proc.longlength	proc.azlength	proc.udlonglength	proc.sslonglength
proc.decstep	proc.decstep	proc.latstep	proc.elevstep	proc.udlatstep	proc.sslatstep
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.decoffset	proc.decoffset	proc.latoffset	proc.elevoffset	proc.udlatoffset	proc.sslatoffset
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.startnumber					
proc.repeatnumber					
proc.offinterval					
proc.offduration					
proc.backandforth					
proc.realtimedisplay					
proc.data					

Table 3: Keywords available for the ralongmap procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

4.1.7 offonoff procedure

This procedure tracks a fixed point in the chosen coordinate system or moves the telescope from that point at a constant rate in one or both coordinates. The available keywords for the offonoff procedure are shown in Table 6.

4.1.8 fivepoint procedure

Five-Point steps the telescope through the sequence off on +maj -maj +min -min on off, where maj is the R.A. or Longitude Offset, and min is the Dec., latitude, or elevation Offset. This procedures primary purpose is to determine pointing offsets. If you are observing in Azimuth/Elevation coordinates, the blank sky off position will be 5 times the Az. Offset distance from the central position. Otherwise, an additional Az. Offset parameter is specified. Point Integration is the integration time on each position. Unless Separate Scans = no, each integration will be recorded as a separate scan. The available keywords for the fivepoint procedure are shown in Table 7.

4.1.9 circle procedure

Circle moves the telescope in a circle with the specified Radius around the given position in the chosen coordinate system. The available keywords for the circle procedure are shown in Table 8.

4.1.10 tipping procedure

Tipping drives the telescope in elevation at a fixed azimuth to determine atmospheric attenuation. The direction of elevation motion is determined by Start and Stop elevations. Any sign on the Elevation Rate will be ignored. The available keywords for the tipping procedure are shown in Table 9.

4.1.11 focusprime procedure

Focus Prime scans the prime focus receiver in its feeds axial direction. The primary purpose is to determine the receivers maximum gain focus position. The available keywords for the focusprime procedure are shown in Table 10.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.decrate	proc.decrate	proc.latrate	proc.elevrate	proc.udlatrate	proc.sslatrate
proc.declength	proc.declength	proc.latlength	proc.elevlength	proc.udlatlength	proc.sslatlength
proc.rastep	proc.hastep	proc.longstep	proc.azstep	proc.udlongstep	proc.sslongstep
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.decoffset	proc.decoffset	proc.latoffset	proc.elevoffset	proc.udlatoffset	proc.sslatoffset
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.startnumber					
proc.repeatnumber					
proc.offinterval					
proc.offduration					
proc.backandforth					
proc.realtimedisplay					
proc.data					

Table 4: Keywords available for the declatmap procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

4.1.12 peak procedure

The Peak procedure sweeps through the specified position in the chosen coordinate system in the four cardinal directions. Its primary purpose to deterime pointing offsets. Each sweep will be a separate scan with the Scan Length automatically determined by the coordinate Lengths and Rates. After the sweep along the major axis pointing corrections are expected to be entered in the pop-up window. After the pop-up window is closed the minor axis pointing will be done. This ensures(?) that the minor axis pointing will include the peak of the source flux. The available keywords for the peak procedure are shown in Table 11.

4.1.13 pointfocussubreflector procedure

Point Focus Subreflector does a cross pointing, moves the subflector and then repeats itself. The primary purpose is to determine the subreflectors maximum gain focus position. The available keywords for the pointfocussubreflector procedure are shown in Table 12.

4.1.14 slewto pseudo-procedure

SlewTo moves the telescope to the commanded position. The available keywords for the slewto procedure are shown in Table 13.

4.1.15 offon procedure

Off-On does two data integrations in an off-on-source sequence. The on-source position is at the specified coordinates. The off-source and on-source integration times are the same. The available keywords for the offon procedure are shown in Table 14.

4.1.16 onoff procedure

On-Off does two data integrations in an on-off-source sequence. The on-source position is at the specified coordinates. The off-source and on-source integration times are the same. The available keywords for the onoff procedure are shown in Table 15.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.rastep	proc.hastep	proc.longstep	proc.azstep	proc.udlongstep	proc.sslongstep
proc.decstep	proc.decstep	proc.latstep	proc.elevstep	proc.udlatstep	proc.sslatstep
proc.rapoints	proc.hapoints	proc.longpoints	proc.azpoints	proc.udlongpoints	proc.sslongpoints
proc.decpoints	proc.decpoints	proc.latpoints	proc.elevpoints	proc.udlatpoints	proc.sslatpoints
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.decoffset	proc.decoffset	proc.latoffset	proc.elevoffset	proc.udlatoffset	proc.sslatoffset
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.startnumber					
proc.repeatnumber					
proc.offinterval					
proc.pointduration					
proc.realtimedisplay					
proc.data					

Table 5: Keywords available for the pointmap procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.decoffset	proc.decoffset	proc.latoffset	proc.elevoffset	proc.udlatoffset	proc.sslatoffset
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.repeatnumber					
proc.onduration					
proc.realtimedisplay					
proc.data					

Table 6: Keywords available for the offonoff procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

4.1.17 nod procedure

Nod does two data integrations in an one 1st beam on 2nd beam sequence. The 1st beam and the 2nd beam are defined by the user. The times spent observing on each beam is the same. The available keywords for the nod procedure are shown in Table 16.

4.1.18 focussubreflectorprocedure

The FocusSubreflector procedure definition. It scans the gregorian subreflector along its Y (axial) direction. The primary purpose is to determine the receivers maximum gain as a function of the gregorian subreflectors axial focus position. The available keywords for the focussubreflector procedure are shown in Table 17.

4.1.19 List of all procedure keywords

- proc.startutc
- proc.startlst

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.decoffset	proc.decoffset	proc.latoffset	proc.elevoffset	proc.udlatoffset	proc.sslatoffset
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.repeatnumber					
proc.startnumber					
proc.pointduration					
proc.realtimedisplay					
proc.data					

Table 7: Keywords available for the fivepoint procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.repeatnumber					
proc.radius					
proc.startangle					
proc.angularrate					
proc.realtimedisplay					
proc.data					

Table 8: Keywords available for the circle procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

- proc.stoputc
- proc.stoplst
- proc.scanduration
- proc.symmetric
- proc.onbeam
- proc.startnumber
- proc.repeatnumber
- proc.pointduration
- proc.onduration
- proc.offduration
- proc.offinterval
- proc.backandforth
- proc.separatescans
- proc.data
- proc.numsweeps

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.repeatnumber					
proc.azimuth					
proc.startelev					
proc.stopelev					
proc.realtimedisplay					
proc.data					

Table 9: Keywords available for the tipping procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.repeatnumber					
proc.startfocus					
proc.stopfocus					
proc.flocusrate					
proc.realtimedisplay					
proc.data					

Table 10: Keywords available for the focusprime procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

- proc.radius
- proc.startangle
- proc.angularrate
- proc.startelev
- proc.stopelev
- proc.startfocus
- proc.stopfocus
- proc.focusrate
- proc.startrotation
- proc.stoprotation
- proc.rotationrate
- proc.parm1
- proc.parm2
- proc.parm3
- proc.parm4
- proc.parm5
- proc.parm6

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.rarate	proc.harate	proc.longrate	proc.azrate	proc.udlongrate	proc.sslongrate
proc.decrate	proc.decrate	proc.latrate	proc.elevrate	proc.udlatrate	proc.sslatrate
proc.ralength	proc.halength	proc.longlength	proc.azlength	proc.udlonglength	proc.sslonglength
proc.declength	proc.declength	proc.latlength	proc.elevlength	proc.udlatlength	proc.sslatlength
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.startnumber					
proc.repeatnumber					
proc.realtimedisplay					
proc.autoupdatelpc					
proc.data					

Table 11: Keywords available for the peak procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

- proc.parm7
- proc.parm8
- proc.parm9
- proc.ra
- proc.dec
- proc.rarate
- proc.decrate
- proc.ralength
- proc.declength
- proc.rastep
- proc.decstep
- proc.rapoints
- proc.decpoints
- proc.raoffset
- proc.decoffset
- proc.secantdec
- proc.ha
- proc.harate
- proc.halength
- proc.hastep
- proc.hapoints
- proc.haoffset

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.rarate	proc.harate	proc.longrate	proc.azrate	proc.udlongrate	proc.sslongrate
proc.decrate	proc.decrate	proc.latrate	proc.elevrate	proc.udlatrate	proc.sslatrate
proc.ralength	proc.halength	proc.longlength	proc.azlength	proc.udlonglength	proc.sslonglength
proc.declength	proc.declength	proc.latlength	proc.elevlength	proc.udlatlength	proc.sslatlength
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.subreflectorxstart					
proc.subreflectorxstop					
proc.subreflectorxstep					
proc.subreflectorystart					
proc.subreflectorystop					
proc.subreflectorystep					
proc.subreflectorzstart					
proc.subreflectorzstop					
proc.subreflectorzstep					
proc.startnumber					
proc.repeatnumber					
proc.realtimedisplay					
proc.autoupdatelpc					
proc.data					

Table 12: Keywords available for the pointfocussubreflector procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.data					

Table 13: Keywords available for the slewto procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

- proc.long
- proc.lat
- proc.longrate
- proc.latrate
- proc.longlength
- proc.latlength
- proc.longstep
- proc.latstep
- proc.longpoints
- proc.latpoints
- proc.longoffset

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.decoffset	proc.decoffset	proc.latoffset	proc.elevoffset	proc.udlatoffset	proc.sslatoffset
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.repeatnumber					
proc.onduration					
proc.realtimedisplay					
proc.data					

Table 14: Keywords available for the offon procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.raoffset	proc.haoffset	proc.longoffset	proc.azoffset	proc.udlongoffset	proc.sslongoffset
proc.decoffset	proc.decoffset	proc.latoffset	proc.elevoffset	proc.udlatoffset	proc.sslatoffset
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.repeatnumber					
proc.onduration					
proc.realtimedisplay					
proc.data					

Table 15: Keywords available for the onoff procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

- proc.latoffset
- proc.secantlat
- proc.az
- proc.elev
- proc.azrate
- proc.elevrate
- proc.azlength
- proc.elevlength
- proc.azstep
- proc.elevstep
- proc.azpoints
- proc.elevpoints
- proc.azoffset
- proc.elevoffset
- proc.secantelev

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.rarate	proc.harate	proc.longrate	proc.azrate	proc.udlongrate	proc.sslongrate
proc.decrate	proc.decrate	proc.latrate	proc.elevrate	proc.udlatrate	proc.sslatrate
proc.secantdec	proc.secantdec	proc.secantlat	proc.secantelev	proc.secantudlat	proc.secantsslat
proc.onbeam					
proc.symmetric					
proc.repeatnumber					
proc.scanduration					
proc.realtimedisplay					
proc.data					

Table 16: Keywords available for the nod procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

RA & DEC	HA & DEC	Galactic	Az. & El. / Encoder	User Defined	Solar System
proc.ra	proc.ha	proc.long	proc.az	proc.udlong	proc.sslong
proc.dec	proc.dec	proc.lat	proc.elev	proc.udlat	proc.sslat
proc.subreflectorystart					
proc.subreflectorystop					
proc.repeatnumber					
proc.focusrate					
proc.realtimedisplay					
proc.autoupdatefocus					
proc.data					

Table 17: Keywords available for the focussubreflector procedure. The keywords that are active depends on the selection of the sc.coordinatemode and sc.offsetcoordinatemode keyword values.

- proc.udlong
- proc.udlat
- proc.udlongrate
- proc.udlatrate
- proc.udlonglength
- proc.udlatlength
- proc.udlongstep
- proc.udlatstep
- proc.udlongpoints
- proc.udlatpoints
- proc.udlongoffset
- proc.udlatoffset
- proc.secantudlat
- proc.sslong

- proc.sslat
- proc.sslongrate
- proc.sslatrate
- proc.sslonglength
- proc.sslatlength
- proc.sslongstep
- proc.sslatstep
- proc.sslongpoints
- proc.sslatpoints
- proc.sslongoffset
- proc.sslatoffset
- proc.secantsslat
- proc.autoupdatelpc
- proc.autoupdatefocus
- proc.realtimedisplay
- proc.primefocusaxial
- proc.primefocusaxialrate
- proc.primefocusrotation
- proc.primefocusrotationrate
- proc.primefocustranslation
- proc.primefocustranslationrate
- proc.subreflectorx
- proc.subreflectorxrate
- proc.subreflectorxstart
- proc.subreflectorxstop
- proc.subreflectorxstep
- proc.subreflectory
- proc.subreflectoryrate
- proc.subreflectorystart
- proc.subreflectorystop
- proc.subreflectorystep
- proc.subreflectorz
- proc.subreflectorzrate

- proc.subreflectorzstart
- proc.subreflectorzstop
- proc.subreflectorzstep
- proc.subreflectoracty1
- proc.subreflectoracty1rate
- proc.subreflectoracty2
- proc.subreflectoracty2rate
- proc.subreflectoracty3
- proc.subreflectoracty3rate
- proc.subreflectoractx1
- proc.subreflectoractx1rate
- proc.subreflectoractx2
- proc.subreflectoractx2rate
- proc.subreflectoractz1
- proc.subreflectoractz1rate
- proc.offsetfocus1x
- proc.offsetfocus1xrate
- proc.offsetfocus1y
- proc.offsetfocus1yrate
- proc.offsetfocus1z
- proc.offsetfocus1zrate
- proc.offsetfocus2x
- proc.offsetfocus2xrate
- proc.offsetfocus2y
- proc.offsetfocus2yrate
- proc.offsetfocus2z
- proc.offsetfocus2zrate
- proc.neworigin
- proc.originlongitude
- proc.originlongvel
- proc.originlatitude
- proc.originlatvel
- proc.originrotation

- proc.originrotvel
- proc.originstartutc
- proc.originstartlst
- proc.orbitrefframe
- proc.orbitepoch
- proc.pericenterepoch
- proc.meananomaly
- proc.semimajoraxis
- proc.orbitperiod
- proc.meandailymotion
- proc.eccentricity
- proc.pericenterdistance
- proc.pericenterargument
- proc.longascendingnode
- proc.orbitinclination

The parameters in the procedure group are defined as they apply to built-in observing procedures. You may use them for other purposes. If you change a parameter value, the value will be changed for all procedures that use it. At the end of this list are nine generic parameters called parm1-9. These are not used by any built-in procedures. You may assign labels and units to these parameters when you attach your procedure to the the interactive part of GBT Observe.

For most observations procedures will be executed by the GBT as soon as possible, but the start time may be explicitly specified in either UTC or LST with one of the following two parameters. If both start_utc and start_lst are specifed, all but the last one set will be ignored.

proc.startutc

Description: Start UTC is a parameter that allows you to start a procedure at a specified Universal Coordinated Time rather than just as soon as possible. The time is assumed to be within the day starting 1/2 hour before and ending 23 1/2 hours after the current time. The start time is valid for one scan only and is cleared after a scan is initiated. Procedures that run more than one scan will start all scans after the first one a.s.a.p. Subsequent procedure will also start a.s.a.p. unless a new start time is specified before each is invoked. This parameter is assigned as a string in sexagesimal format, 'HH:MM:SS.s'

Units: HH:MM:SS.S

Example of Use: proc.startutc = "03:45:34"

proc.startlst

Description: Start LST is a parameter that allows you to start a procedure at a specified Local Apparent Sidereal Time rather than just as soon as possible. The time is assumed to be within the day starting 1/2 hour before and ending 23 1/2 hours after the current time. The start time is valid for one scan only and is cleared after a scan is initiated. Procedures that run more than one scan will start all scans after the first one a.s.a.p. Subsequent procedure will also start a.s.a.p. unless a new start time is specified before each is invoked. This parameter is assigned as a string in sexagesimal format, 'HH:MM:SS.s'

Units: HH:MM:SS.S

Example of Use: proc.startlst = "00:00:01"

The "track" procedure has three other timing parameters that are unique to it. A stop UTC or LST may be specified so that the scan will be started as soon as possible, but the scan is guaranteed to end at the specified time. This is useful for VLBI observations to keep schedules synchronized at different observatories. When a stop time is used the length of the scan will depend on when the telescope is able to get on position and initiate the data-taking process. If both stop_utc and stop_lst are specifed, all but the last one set will be ignored.

If a start time or no start or stop time is given, the length of a "track" scan is set by the scan_duration parameter. All procedure start and stop times are cleared after a procedure is executed so that the next procedure will be executed as soon as possible unless a new time is specified.

proc.stoputc

Description: Stop UTC is a parameter that allows you to stop the first scan of a procedure at a specified Universal Coordinated Time. The scan is started as soon as possible. The Stop UTC is generally useful with the 'track' procedure where one wants to start tracking an object as soon as possible while being guaranteed that the telescope will move to the next source at the specified time. This is often useful for VLBI observations. The time is assumed to be within the day starting 1/2 hour before and ending 23 1/2 hours after the current time. The stop time is valid for one scan only and is cleared after a scan is initiated. This parameter is assigned as a string in sexagesimal format, 'HH:MM:SS.s'

Units: HH:MM:SS.S

Example of Use: proc.stoputc = "23:59:59"

proc.stoplst

Description: Stop LST is a parameter that allows you to stop the 'track' procedure at a specified Local Apparent Sidereal Time. The scan is started as soon as possible. The Stop LST is generally useful when one wants to start tracking an object as soon as possible while being guaranteed that the telescope will move to the next source at the specified time. This is often useful for VLBI observations. The time is assumed to be within the day starting 1/2 hour before and ending 23 1/2 hours after the current time. The stop time is valid for one scan only and is cleared after a scan is initiated. This parameter is assigned as a string in sexagesimal format, 'HH:MM:SS.s'

Units: HH:MM:SS.S

Example of Use: proc.stoplst = "09:45:12"

proc.scanduration

Description: Scan Duration is the length of the 'track' procedure scan in UTC seconds. Any data integrations completed after the end of a scan will normally be discarded. Hence, the Scan Duration is typically an integer number of integration times plus a second or two. This parameter is ignored if a stop time is specified.

Values: integer

Allowed Range: $0 \leq$ proc.scanduration $\leq \infty$

Units: seconds

Example of Use: proc.scanduration = 60

For the "Nod" procedure you have to define how the telescope will nod between the two beams and which beam is the "reference" beam and which beam is the "signal" beam. This is done using the proc.symettric and proc.onbeam parameter.

proc.symmetric

Description: If symmetric is set to yes then the nod procedure observes the 1st beam, 2nd beam, 2nd beam, 1st beam using four scans. If symmetric is set to no then the nod procedure observes the 1st beam and then the 2nd beam using two scans.

Values: YES, NO

Units: N/A

Example of Use: proc.symmetric = NO

proc.onbeam

Description: onbeam specifies which beam is to be used for the first scan in the nod procedure

Values: "1", "2", "3", "4"

Units: N/A

Example of Use: proc.onbeam = "1"

When mapping GO has to define various start and stop scans as well as duration for each position as well as offsets and grid sizes.

proc.startnumber

Description: Start Number allows you to resume a partially completed Cross, raster map, or Point-Map by specifying the sweep or point number to start with.

Values: integer

Allowed Range: $0 \leq$ proc.startnumber $\leq \infty$

Units: N/A

Example of Use: proc.startnumber = 3

proc.repeatnumber

Description: Repeats specifies the number of times the procedure is to be repeated with the same parameter values.

Values: integer

Allowed Range: $0 \leq$ proc.repeatnumber $\leq \infty$

Units: N/A

Example of Use: proc.repeatnumber = 3

proc.pointduration

Description: Point Integration is the integration time, in seconds, spent on each location of a map grid or a five-point pointing procedure.

Values: float

Allowed Range: $0 \leq$ proc.pointduration $\leq \infty$

Units: seconds

Example of Use: proc.pointduration = 10

proc.onduration

Description: The On Integration is the integration time, in seconds, spent on each location of an off-on-off observing procedure.

Values: float

Allowed Range: $0 \leq$ proc.onduration $\leq \infty$

Units: seconds

Example of Use: proc.onduration = 60

proc.offduration

Description: The Off Integration is the integration time, in seconds, spent on each off observation.

Values: float

Allowed Range: $0 \leq$ proc.offduration $\leq \infty$

Units: seconds

Example of Use: proc.offduration = 20

proc.offinterval

Description: Off Interval specifies the number of locations in a Point-Map that are integrated before making an ”off” integration outside the map area. For a RALongMap or DecLatMap Off Interval specifies the number of rows of the map to be made between ”off” observations.

Values: float

Allowed Range: $0 \leq$ proc.offinterval $\leq \infty$

Units: seconds

Example of Use: proc.offinterval = 4

proc.backandforth

Description: Back&Forth selects whether or not alternate sweeps in a raster scan map are traced in opposite directions. By selecting ”yes” you save time by not having the telescope retrace to the same side of the map for each sweep.

Values: YES, NO

Units: N/A

Example of Use: proc.backandforth = YES

proc.separatescans

Description: Separate Scans specifies whether each integration in an on-off or five-point procedure is recorded as a separate scan with different scan numbers.

Values: YES, NO

Units: N/A

Example of Use: proc.separatescans = YES

proc.data

Description: The Record Data button selects whether data are recorded during a scan. The data might be temporarily turned off to make a trial run of a procedure to test the telescope motions. This is a spring-loaded value in the sense that it returns to ”on” at the end of each scan.

Values: YES, NO

Units: N/A

Example of Use: proc.data = YES

proc.numsweeps

Description: Number of Sweeps specifies the number of sweeps in a raster map. An odd number of sweeps will run the center sweep through the specified map center position. An equal number of sweeps will be made on either side of the center position.

Values: integer

Allowed Range: $0 \leq$ proc.numsweeps $\leq \infty$

Units: N/A

Example of Use: proc.numsweeps = 11

proc.radius

Description: This parameter specifies the Radius, in arcminutes, of the circle traced by the Circle procedure

Values: float

Allowed Range: $0 \leq$ proc.radius $\leq \infty$

Units: arc minutes

Example of Use: proc.radius = 5.3

proc.startangle

Description: Start Angle specifies the parallactic angle, in degrees, for the beginning of the trace in the Circle procedure. Zero degrees is north; 90 degrees is east.

Values: float

Units: degrees

Example of Use: proc.startangle = 45.0

proc.angularrate

Description: Angular Rate specifies the rate of scan, in degrees per minute, along the circle in the Circle procedure.

Values: float

Units: degrees per minute

Example of Use: proc.angularrate = 12.4

proc.startelev

Description: This parameter specifies the Start Elevation, in degrees, of a Tipping scan. The start and stop elevations determine the direction of scan, and the sign of the scan rate is ignored.

Values: float

Allowed Range: $5.0 \leq$ proc.startelev ≤ 90.0

Units: degrees

Example of Use: proc.startelev = 85.0

proc.stopelev

Description: This parameter specifies the Stop Elevation, in degrees, of a Tipping scan. The start and stop elevations determine the direction of scan, and the sign of the scan rate is ignored.

Values: float

Allowed Range: $5.0 \leq$ proc.stopelev ≤ 90.0

Units: degrees

Example of Use: proc.stopelev = 9.56

proc.startfocus

Description: This parameter specifies the Start Focus position, in millimeters, of a Focus scan. The start and stop positions determine the direction of scan, and the sign of the Focus Rate is ignored.

Values: float

Allowed Range: $-575 \leq$ proc.startfocus ≤ 290

Units: millimeters

Example of Use: proc.startfocus = -9.45

proc.stopfocus

Description: This parameter specifies the Stop Focus position, in millimeters, of a Focus Prime procedure. The start and stop positions determine the direction of scan, and the sign of the Focus Rate is ignored.

Values: float

Allowed Range: $-575 \leq$ proc.stopfocus ≤ 290

Units: millimeters

Example of Use: proc.stopfocus = 60.0

proc.focusrate

Description: This parameter is the Focus Rate, in millimeters per minute, of the prime focus receiver in the Focus Prime procedure. The start and stop positions determine the direction of scan, and the sign of the Focus Rate is ignored.

Values: float

Allowed Range: $0 \leq$ proc.focusrate $\leq \infty$

Units: mm per second

Example of Use: proc.focusrate = 30.0

proc.startrotation

Description: This parameter specifies the Start Rotation position, in degrees, of a prime focus receiver rotation scan.

Values: float

Allowed Range: $0.0 \leq$ proc.startrotation ≤ 360.0

Units: degrees

Example of Use: proc.startrotation = 10.9

proc.stoprotation

Description: This parameter specifies the Stop Rotation position, in degrees, of a prime focus receiver rotation scan.

Values: float

Allowed Range: $0.0 \leq$ proc.stoprotation $\leq 360.$

Units: degrees

Example of Use: proc.stoprotation = 256.6

proc.rotationrate

Description: This parameter is the Rotation Rate, in degrees per minute, of the prime focus receiver.

Values: float

Allowed Range: $0.0 \leq$ proc.rotationrate $\leq \infty$

Units: degrees per minute

Example of Use: proc.rotationrate = 22.5

proc.parm1

Description: This parameter, parm1, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm1 = YES

proc.parm2

Description: This parameter, parm2, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm2 = ”4.5”

proc.parm3

Description: This parameter, parm3, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm3 = ”Pluto”

proc.parm4

Description: This parameter, parm4, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm4 = ”-2.9”

proc.parm5

Description: This parameter, parm5, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm5 = NO

proc.parm6

Description: This parameter, parm6, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm6 = ”T”

proc.parm7

Description: This parameter, parm7, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm7 = ”F”

proc.parm8

Description: This parameter, parm8, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm8 = ”Einstein”

proc.parm9

Description: This parameter, parm9, is one of nine generic string value parameters that may be used in writing new procedures in glish.

Values: any string

Units: N/A

Example of Use: proc.parm9 = ”01:56:34”

Procedures may be written to handle coordinates in any of the provided coordinate systems using generic coordinates, rather than specific coordinates such as ra, dec, az, elev, etc. See the description of these above. Right Ascension and Declination coordinates. The epoch of this coordinate is set by the Coordinate Mode as J2000, B1950, or Current RA/Dec.

proc.ra

Description: R.A. is the right ascension, in HH:MM:SS.SS format, at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Units: HH:MM:SS.S

Example of Use: proc.ra = ”12:45:56.789”

proc.dec

Description: Dec. is the declination, in sDD:MM:SS.S format, at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Units: HH:MM:SS.S

Example of Use: proc.dec = ”-05:56:12.76”

proc.rarate

Description: R.A. Rate is the right ascension rate, in arcminutes per minute, of the telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.rarate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.rarate = 3.0

proc.decrate

Description: Dec. Rate is the declination rate, in arcminutes per minute, of the telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.decrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.decrate = 5.6

proc.ralength

Description: R.A. Length is the full length, in arcminutes, of a right ascension sweep in a Cross or map procedure. The sweep is centered on the specified R.A. coordinate.

Values: float

Units: arcminutes

Example of Use: proc.ralength = 34.0

proc.declength

Description: Dec. Length is the full length, in arcminutes, of a declination sweep in a Cross or map procedure. The sweep is centered on the specified Dec. coordinate.

Values: float

Units: arcminutes

Example of Use: proc.declength = 23.5

proc.rastep

Description: R.A. Step is the right ascension step size, in arcminutes, between declination sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or rows of point locations are centered on the specified R.A. coordinate.

Values: float

Allowed Range: $0.0 \leq$ proc.rastep $\leq \infty$

Units: arcminutes

Example of Use: proc.rastep = 1.0

proc.decstep

Description: Dec. Step is the declination step size, in arcminutes, between right ascension or hour angle sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or columns of point locations are centered on the specified Dec. coordinate.

Values: float

Allowed Range: float ¿ $0.0 \leq$ proc.decstep ≤ 0.0

Units: ∞

Example of Use: arcminutes

proc.rapoints

Description: Number of R.A. Points specifies the number of Point-Map procedure locations in the right ascension coordinate. An odd number will put the center column on the specified R.A. coordinate.

Values: integer

Units: N/A

Example of Use: proc.rapoints = 23

proc.decpoints

Description: Number of Dec. Points specifies the number of Point-Map procedure locations in the declination coordinate. An odd number will put the center row on the specified Dec. coordinate.

Values: integer

Units: N/A

Example of Use: proc.decpoints = 3

proc.raoffset

Description: R.A. Offset is the right ascension offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or for two of the locations in a Five-Point procedure.

Values: float

Allowed Range: $0.0 \leq$ proc.raoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.raoffset = 10.0

proc.decoffset

Description: Dec. Offset is the declination offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or for two of the locations in a Five-Point procedure. The epoch of this coordinate is set by the Coordinate Mode as J2000, B1950, or Current RA/Dec.

Values: float

Allowed Range: $0.0 \leq$ proc.decoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.decoffset = 20.0

proc.secantdec

Description: The secant(dec) selection determines whether right ascension or hour angle offsets and sweep lengths are multiplied by the secant of the declination to determine the actual offsets or lengths on the sky. If secant(dec) is ’no’, the offsets are small circle arc lengths. If secant(dec) is ’yes’, the offsets are great circle arc lengths.

Values: YES, NO

Units: N/A

Example of Use: proc.secantdec = YES

Hour angle coordinates.

proc.ha

Description: H.A. is the hour angle, in HH:MM:SS.SS format, at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Units: HH:MM:SS.S

Example of Use: proc.ha = ”-05:34:56.567”

proc.harate

Description: H.A. Rate is the hour angle rate, in arcminutes per minute, of the telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.harate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.harate = 12.4

proc.halength

Description: H.A. Length is the full length, in arcminutes, of a hour angle sweep in a Cross or map procedure. The sweep is centered on the specified R.A. coordinate.

Values: float

Allowed Range: $0.0 \leq$ proc.halength $\leq \infty$

Units: arminutes

Example of Use: proc.halength = 123.9

proc.hastep

Description: H.A. Step is the hour angle step size, in arcminutes, between declination sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or rows of point locations are centered on the specified H.A. coordinate.

Values: float

Allowed Range: $0.0 \leq$ proc.hastep $\leq \infty$

Units: arcminutes

Example of Use: proc.hastep = 1.5

proc.hapoints

Description: Number of H.A. Points specifies the number of Point-Map procedure locations in the hour angle coordinate. An odd number will put the center column on the specified H.A. coordinate.

Values: integer

Units: N/A

Example of Use: proc.hapoints = 32

proc.haoffset

Description: H.A. Offset is the hour angle offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or for two of the locations in a Five-Point procedure.

Values: float

Allowed Range: $0.0 \leq$ proc.haoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.haoffset = 15.0

Galactic Longitude and Latitude coordinates.

proc.long

Description: This parameter is the galactic Longitude, in decimal degrees, at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Values: float

Allowed Range: $0.0 \leq$ proc.long ≤ 360.0

Units: degrees

Example of Use: proc.long = 28.07

proc.lat

Description: This parameter is the galactic Latitude, in decimal degrees, at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Values: float

Allowed Range: $-90.0 \leq$ proc.lat ≤ 90.0

Units: degrees

Example of Use: proc.lat = 0.05

proc.longrate

Description: This parameter is the galactic Longitude Rate, in arcminutes per minute, of the telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.longrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.longrate = 1.2

proc.latrate

Description: This parameter is the galactic Latitude Rate, in arcminutes per minute, of the telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.latrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.latrate = 4.5

proc.longlength

Description: Long. Length is the full length, in arcminutes, of a galactic longitude sweep in a Cross or map procedure. The sweep is centered on the specified longitude coordinate.

Values: float

Allowed Range: $0 \leq$ proc.longlength $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.longlength = 45.7

proc.latlength

Description: Lat. Length is the full length, in arcminutes, of a galactic latitude sweep in a Cross or map procedure. The sweep is centered on the specified latitude coordinate.

Values: float

Allowed Range: $0 \leq$ proc.latlength $\leq \infty$

Units: arcminutes

Example of Use: proc.latlength = 34.2

proc.longstep

Description: Long. Step is the galactic longitude step size, in arcminutes, between latitude sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or rows of point locations are centered on the specified longitude coordinate.

Values: float

Allowed Range: $0 \leq$ proc.longstep $\leq \infty$

Units: arcminutes

Example of Use: proc.longstep = 2.5

proc.latstep

Description: Lat. Step is the galactic latitude step size, in arcminutes, between longitude sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or columns of point locations are centered on the specified latitude coordinate.

Values: float

Allowed Range: $0 \leq$ proc.latstep $\leq \infty$

Units: arcminutes

Example of Use: proc.latstep = 3.0

proc.longpoints

Description: Number of Lon. Points specifies the number of Point-Map procedure locations in the galactic longitude coordinate. An odd number will put the center column on the specified longitude coordinate.

Values: integer

Units: N/A

Example of Use: proc.longpoints = 5

proc.latpoints

Description: Number of Lat. Points specifies the number of Point-Map procedure locations in the galactic latitude coordinate. An odd number will put the center row on the specified latitude coordinate.

Values: integer

Units: N/A

Example of Use: proc.latpoints = 10

proc.longoffset

Description: Long. Offset is the galactic longitude offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or two of the locations in a Five-Point procedure.

Values: float

Allowed Range: $0 \leq$ proc.longoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.longoffset = 25.0

proc.latoffset

Description: Lat. Offset is the galactic latitude offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or two of the locations in a Five-Point procedure.

Values: float

Allowed Range: $0 \leq$ proc.latoffset $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.latoffset = 30.0

proc.secantlat

Description: The secant(lat) selection determines whether galactic longitude offsets and sweep lengths are multiplied by the secant of the latitude to determine the actual offsets or lengths on the sky. If secant(lat) is ’no’, the offsets are small circle arc lengths. If secant(lat) is ’yes’, the offsets are great circle arc lengths.

Values: YES, NO

Units: N/A

Example of Use: proc.secantlat = YES

Azimuth, Elevation coordinates.

proc.az

Description: This parameter is the Azimuth, in decimal degrees, at the beginning of the Track procedure or the center of a map, Cross, or Circle. Azimuth is 0.0 degrees at north and +90.0 degrees at east.

Values: float

Allowed Range: $0.0 \leq$ proc.az ≤ 360.0

Units: degrees

Example of Use: proc.az = 234.56

proc.elev

Description: This parameter is the Elevation, in decimal degrees, at the beginning of the Track procedure or the center of a map, Cross, or Circle. The minimum elevation of the GBT is 5 degrees.

Values: float

Allowed Range: $5.0 \leq$ proc.elev ≤ 90.0

Units: degrees

Example of Use: proc.elev = 23.67

proc.azrate

Description: This parameter is the Azimuth Rate, in arcminutes per minute, of the telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.azrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.azrate = 13.2

proc.elevrate

Description: This parameter is the Elevation Rate, in arcminutes per minute, of the telescope motion in the Track, map, Cross, or Tipping procedure.

Values: float

Allowed Range: $0 \leq$ proc.elevrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.elevrate = 12.5

proc.azlength

Description: Az. Length is the full length, in arcminutes, of an azimuth sweep in a Cross or map procedure. The sweep is centered on the specified azimuth coordinate.

Values: float

Allowed Range: $0 \leq$ proc.azlength $\leq \infty$

Units: arcminutes

Example of Use: proc.azlength = 10.0

proc.elevlength

Description: Elev. Length is the full length, in arcminutes, of an elevation sweep in a Cross or map procedure. The sweep is centered on the specified elevation coordinate.

Values: float

Allowed Range: $0 \leq$ proc.elevlength $\leq \infty$

Units: arcminutes

Example of Use: proc.elevlength = 45.0

proc.azstep

Description: Az. Step is the azimuth step size, in arcminutes, between elevation sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or rows of point locations are centered on the specified azimuth coordinate.

Values: float

Allowed Range: $0 \leq$ proc.azstep $\leq \infty$

Units: arcminutes

Example of Use: proc.azstep = 3.5

proc.elevstep

Description: Elev. Step is the elevation step size, in arcminutes, between azimuth sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or columns of point locations are centered on the specified elevation coordinate.

Values: float

Allowed Range: $0 \leq$ proc.elevstep $\leq \infty$

Units: arcminutes

Example of Use: proc.elevstep = 0.5

proc.azpoints

Description: Number of Az. Points specifies the number of Point-Map procedure locations in the azimuth coordinate. An odd number will put the center column on the specified azimuth coordinate.

Values: integer

Units: N/A

Example of Use: proc.azpoints = 10

proc.elevpoints

Description: Number of El. Points specifies the number of Point-Map procedure locations in the elevation coordinate. An odd number will put the center row on the specified elevation coordinate.

Values: integer

Units: N/A

Example of Use: proc.elevpoints = 15

proc.azoffset

Description: Az. Offset is the azimuth offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or two of the locations in a Five-Point procedure. It is also used as an azimuth offset in the Five-Point procedure when working in a coordinate system other than Azimuth/Elevation.

Values: float

Allowed Range: $0 \leq$ proc.azoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.azoffset = 35.0

proc.elevoffset

Description: Elev. Offset is the elevation offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or two of the locations in a Five-Point procedure.

Values: float

Allowed Range: $0 \leq$ proc.elevoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.elevoffset = 40.0

proc.secantelev

Description: The secant(elev) selection determines whether azimuth offsets and sweep lengths are multiplied by the secant of the elevation to determine the actual offsets or lengths on the sky. If secant(elev) is ’no’, the offsets are small circle arc lengths. If secant(elev) is ’yes’, the offsets are great circle arc lengths.

Values: YES, NO

Units: N/A

Example of Use: proc.secantelev = NO

User-Defined coordinates. These coordinates are defined by the user by specifying the location of a spherical coordinate pole and prime meridian in J2000 coordinates, and, optionally, the first and second time derivatives of these locations.

proc.udlong

Description: This parameter is the Longitude in ”User Defined” coordinates, in decimal degrees, at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Values: float

Allowed Range: $0.0 \leq$ proc.udlong ≤ 360.0

Units: degrees

Example of Use: proc.udlong = 20.5

proc.udlat

Description: This parameter is the Latitude in ”User Defined” coordinates, in decimal degrees, at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Values: float

Allowed Range: -90.0 $\leq$ proc.udlat $\leq$ +90.0

Units: degrees

Example of Use: proc.udlat = -14.6

proc.udlongrate

Description: This parameter is the Longitude Rate, in arcminutes per minute, in ”User Defined” spherical coordinates. This is the rate of telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: 0 $\leq$ proc.udlongrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.udlongrate = 56.4

proc.udlatrate

Description: This parameter is the Latitude Rate, in arcminutes per minute, in ”User Defined” spherical coordinates. This is the rate of telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: 0 $\leq$ proc.udlatrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.udlatrate = 23.21

proc.udlonglength

Description: Long. Length is the full length, in arcminutes, of a longitude sweep in ”User Defined” coordinates in a Cross or map procedure. The sweep is centered on the specified longitude coordinate.

Values: float

Allowed Range: 0 $\leq$ proc.udlonglength $\leq \infty$

Units: arcminutes

Example of Use: proc.udlonglength = 126.7

proc.udlatlength

Description: Lat. Length is the full length, in arcminutes, of a latitude sweep in ”User Defined” coordinates in a Cross or map procedure. The sweep is centered on the specified latitude coordinate.

Values: float

Allowed Range: 0 $\leq$ proc.udlatlength $\leq \infty$

Units: arcminutes

Example of Use: proc.udlatlength = 123.8

proc.udlongstep

Description: Long. Step is the longitude step size, in arcminutes, between latitude sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or rows of point locations are centered on the specified ”User Defined” longitude coordinate.

Values: float

Allowed Range: $0 \leq$ proc.udlongstep $\leq \infty$

Units: arcminutes

Example of Use: proc.udlongstep = 0.1

proc.udlatstep

Description: Lat. Step is the latitude step size, in arcminutes, between longitude sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or columns of point locations are centered on the specified ”User Defined” latitude coordinate.

Values: float

Allowed Range: $0 \leq$ proc.udlatstep $\leq \infty$

Units: arcminutes

Example of Use: proc.udlatstep = 4.0

proc.udlongpoints

Description: Number of Lon. Points specifies the number of Point-Map procedure locations in the ”User Defined” longitude coordinate. An odd number will put the center column on the specified longitude coordinate.

Values: integer

Units: N/A

Example of Use: proc.udlongpoints = 20

proc.udlatpoints

Description: Number of Lat. Points specifies the number of Point-Map procedure locations in the ”User Defined” latitude coordinate. An odd number will put the center row on the specified latitude coordinate.

Values: integer

Units: N/A

Example of Use: proc.udlatpoints = 25

proc.udlongoffset

Description: Long. Offset is the "User Defined" longitude offset, in arcminutes, of the "off" or reference location for the Point-Map or On-Off procedure or for two of the locations in a Five-Point procedure.

Values: float

Allowed Range: $0 \leq$ proc.udlongoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.udlongoffset = 45.0

proc.udlatoffset

Description: Lat. Offset is the "User Defined" latitude offset, in arcminutes, of the "off" or reference location for the Point-Map or On-Off procedure or for two of the locations in a Five-Point procedure.

Values: float

Allowed Range: $0 \leq$ proc.udlatoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.udlatoffset = 50.0

proc.secantudlat

Description: The secant(lat) selection determines whether "User Defined" longitude offsets and sweep lengths are multiplied by the secant of the latitude to determine the actual offsets or lengths on the sky. If secant(lat) is 'no', the offsets are small circle arc lengths. If secant(lat) is 'yes', the offsets are great circle arc lengths.

Values: YES, NO

Units: N/A

Example of Use: proc.secantudlat = NO

Solar System Object coordinates. These coordinates are in a spherical coordinate system whose equator and prime meridian intersection track the location of the selected solar system object (moon, planet, etc.).

proc.sslong

Description: This parameter is the Longitude, in decimal degrees, in spherical coordinates whose equator and prime meridian are tracking a specified solar system object. This longitude is at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Values: float

Allowed Range: $0.0 \leq$ proc.sslong ≤ 360.0

Units: degrees

Example of Use: proc.sslong = 298.0

proc.sslat

Description: This parameter is the Latitude, in decimal degrees, in spherical coordinates whose equator and prime meridian are tracking a specified solar system object. This latitude is at the beginning of the Track procedure or the center of a map, Cross, or Circle.

Values: float

Allowed Range: $-90.0 \leq$ proc.sslat $\leq +90.0$

Units: degrees

Example of Use: proc.sslat = +67.2

proc.sslongrate

Description: This parameter is the Longitude Rate, in arcminutes per minute, in spherical coordinates whose equator and prime meridian are tracking a specified solar system object. This is the rate of telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.sslongrate $\leq \infty$

Units: arcminutes

Example of Use: proc.sslongrate = 1.0

proc.sslatrate

Description: This parameter is the Latitude Rate, in arcminutes per minute, in spherical coordinates whose equator and prime meridian are tracking a specified solar system object. This is the rate of telescope motion in the Track, map, or Cross procedure.

Values: float

Allowed Range: $0 \leq$ proc.sslatrate $\leq \infty$

Units: arcminutes per minute

Example of Use: proc.sslatrate = 12.0

proc.sslonglength

Description: Long. Length is the full length, in arcminutes, of a longitude sweep in spherical coordinates whose equator and prime meridian are tracking a specified solar system object. The sweep is centered on the specified longitude coordinate in a Cross or map procedure.

Values: float

Allowed Range: $0 \leq$ proc.sslonglength $\leq \infty$

Units: arcminutes

Example of Use: proc.sslonglength = 65.3

proc.sslatlength

Description: Lat. Length is the full length, in arcminutes, of a latitude sweep in spherical coordinates whose equator and prime meridian are tracking a specified solar system object. The sweep is centered on the specified latitude coordinate in a Cross or map procedure.

Values: float

Allowed Range: $0 \leq$ proc.sslatlength $\leq \infty$

Units: arcminutes

Example of Use: proc.sslatlength = 79.3

proc.sslongstep

Description: Long. Step is the longitude step size, in arcminutes, between latitude sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or rows of point locations are centered on the specified longitude in spherical coordinates whose equator and prime meridian are tracking a specified solar system object.

Values: float

Allowed Range: $0 \leq$ proc.sslongstep $\leq \infty$

Units: arcminutes

Example of Use: proc.sslongstep = 0.2

proc.sslatstep

Description: Lat. Step is the latitude step size, in arcminutes, between longitude sweeps in a raster map or between locations in a Point-Map procedure. The sweeps or columns of point locations are centered on the specified latitude in spherical coordinates whose equator and prime meridian are tracking a specified solar system object.

Values: float

Allowed Range: $0 \leq$ proc.sslatstep $\leq \infty$

Units: arcminutes

Example of Use: proc.sslatstep = 0.8

proc.sslongpoints

Description: Number of Lon. Points specifies the number of Point-Map procedure locations in the longitude of spherical coordinates whose equator and prime meridian are tracking a specified solar system object. An odd number will put the center column on the specified longitude coordinate.

Values: integer

Units: N/A

Example of Use: proc.sslongpoints = 30

proc.sslatpoints

Description: Number of Lat. Points specifies the number of Point-Map procedure locations in the latitude of spherical coordinates whose equator and prime meridian are tracking a specified solar system object. An odd number will put the center row on the specified latitude coordinate.

Values: integer

Units: N/A

Example of Use: proc.sslatpoints = 35

proc.sslongoffset

Description: Long. Offset is the longitude offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or for two of the locations in a Five-Point procedure. In this case, longitude is defined in spherical coordinates whose equator and prime meridian are tracking a specified solar system object.

Values: float

Allowed Range: $0 \leq$ proc.sslongoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.sslongoffset = 55.0

proc.sslatoffset

Description: Lat. Offset is the latitude offset, in arcminutes, of the ”off” or reference location for the Point-Map or On-Off procedure or for two of the locations in a Five-Point procedure. In this case, latitude is defined in spherical coordinates whose equator and prime meridian are tracking a specified solar system object.

Values: float

Allowed Range: $0 \leq$ proc.sslatoffset $\leq \infty$

Units: arcminutes

Example of Use: proc.sslatoffset = 60.0

proc.secantsslat

Description: The secant(lat) selection determines whether longitude offsets and sweep lengths are multiplied by the secant of the latitude to determine the actual offsets or lengths on the sky. In this case, latitude is defined in spherical coordinates whose equator and prime meridian are tracking a specified solar system object. If secant(lat) is ’no’, the offsets are small circle arc lengths. If secant(lat) is ’yes’, the offsets are great circle arc lengths.

Values: YES, NO

Units: N/A

Example of Use: proc.secantsslat = YES

Real time display control parameters.

proc.autoupdatelpc

Description: This parameter specifies whether automatic updating of the LPC values determined from AIPS++ is to be used or whether the values must be accepted or rejected based on observers reply.

Values: YES, NO

Units: N/A

Example of Use: proc.autoupdatelpc = NO

proc.autoupdatefocus

Description: This parameter specifies whether automatic updating of the subreflector focus Y value determined from AIPS++ is to be used or whether the values must be accepted or rejected based on observers reply.

Values: YES, NO

Units: N/A

Example of Use: proc.autoupdatefocus = YES

proc.realtimedisplay

Description: This parameter specifies which real time display is selected or if the real time display is off.

Values: YES, NO
None, Iards, Python rime

Units: N/A

Example of Use: proc.realtimedisplay = NO
proc.realtimedisplay = Iards

Focus parameters.

proc.primefocusaxial

Description: This parameter is the axial position of the prime focus receiver box in meters. This motion is along a line from roughly the center of the reflecting surface through the prime focus point with positive motion being away from the dish. If the focus tracking mode is on, this position is with respect to the nominally optimum box position for the current elevation. If focus tracking is off, the offset is with respect to the optimum box position at the telescope rigging elevation.

Values: float

Allowed Range: $-\infty \leq$ proc.primefocusaxial $\leq \infty$

Units: meters

Example of Use: proc.primefocusaxial = 0.03

proc.primefocusaxialrate

Description: This parameter is the axial velocity of the prime focus receiver box in meters per second. The box begins its motion at the beginning of a scan at the position specified by the parameter 'primefocusaxial'. Positive motion is away from the dish.

Values: float

Allowed Range: $-\infty \leq$ proc.primefocusaxialrate $\leq \infty$

Units: m/s

Example of Use: proc.primefocusaxialrate = -1.3

proc.primefocusrotation

Description: This parameter is the rotational position of the prime focus receiver box in degrees.

Values: float

Allowed Range: $0.0 \leq$ proc.primefocusrotation ≤ 360.0

Units: degrees

Example of Use: proc.primefocusrotation = 3.7

proc.primefocusrotationrate

Description: This parameter is the rotational velocity of the prime focus receiver box in degrees per second. The box begins its motion at the beginning of a scan at the position specified by the parameter 'primefocusrotation'.

Values: float

Allowed Range: $0.0 \leq$ proc.primefocusrotationrate $\leq \infty$

Units: degrees per second

Example of Use: proc.primefocusrotationrate = 29.4

proc.primefocustranslation

Description: This parameter is the translational offset of the prime focus box in meters. Box translation is in the plane of symmetry of the antenna, perpendicular to the axial focus axis, with positive motion being away from the feed support arm. If the focus tracking mode is on, this offset is with respect to the nominally optimum box position for the current elevation. If focus tracking is off, the offset is with respect to the optimum box position at the telescope rigging elevation.

Values: float

Allowed Range: $0.0 \leq$ proc.primefocustranslation $\leq \infty$

Units: meters

Example of Use: proc.primefocustranslation = 0.2

proc.primefocustranslationrate

Description: This parameter is the translational velocity of the prime focus box in meters per second. The box begins its motion at the beginning of a scan at the position specified by the parameter 'primefocustranslation'. Box translation is in the plane of symmetry of the antenna, perpendicular to the axial focus axis, with positive motion being away from the feed support arm.

Values: float

Allowed Range: $0.0 \leq$ proc.primefocustranslationrate $\leq \infty$

Units: m/s

Example of Use: proc.primefocustranslationrate = 0.95

Subreflector parameters.

proc.subreflectorx

Description: This parameter is the X coordinate offset of the subreflector in meters. If the focus tracking mode is on, this offset is with respect to the nominally optimum subreflector position for the current elevation. If focus tracking is off, the offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorx $\leq \infty$

Units: meters

Example of Use: proc.subreflectorx = 2.3

proc.subreflectorxrate

Description: This parameter is the X coordinate offset velocity of the subreflector in meters per second. The subreflector begins its motion at the beginning of a scan at the position specified by the parameter 'subreflectorx'. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorxrate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectorxrate = 0.4

proc.subreflectorxstart

Description: This parameter is the start position of the X coordinate offset of the subreflector in meters. The offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorxstart $\leq \infty$

Units: meters

Example of Use: proc.subreflectorxstart = -3.0

proc.subreflectorxstop

Description: This parameter is the stop position of the X coordinate offset of the subreflector in meters. The offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorxstop $\leq \infty$

Units: meters

Example of Use: proc.subreflectorxstop = 5.6

proc.subreflectorxstep

Description: This parameter is the X coordinate offset step of the subreflector in the X direction in milliters. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorxstep $\leq \infty$

Units: millimeters

Example of Use: proc.subreflectorxstep = 1.5

proc.subreflectory

Description: This parameter is the Y coordinate offset of the subreflector in meters. If the focus tracking mode is on, this offset is with respect to the nominally optimum subreflector position for the current elevation. If focus tracking is off, the offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectory $\leq \infty$

Units: meters

Example of Use: proc.subreflectory = 3.6

proc.subreflectoryrate

Description: This parameter is the Y coordinate offset velocity of the subreflector in meters per second. The subreflector begins its motion at the beginning of a scan at the position specified by the parameter 'subreflectory'. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoryrate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectoryrate = 0.2

proc.subreflectorystart

Description: This parameter is the start position of the Y coordinate offset of the subreflector in meters. The offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorystart $\leq \infty$

Units: meters

Example of Use: proc.subreflectorystart = 8.3

proc.subreflectorystop

Description: This parameter is the stop position of the Y coordinate offset of the subreflector in meters. The offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorystop $\leq \infty$

Units: meters

Example of Use: proc.subreflectorystop = 4.5

proc.subreflectorystep

Description: This parameter is the Y coordinate offset step of the subreflector in the X direction in milliters. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorystep $\leq \infty$

Units: millimeters

Example of Use: proc.subreflectorystep = 1.9

proc.subreflectorz

Description: This parameter is the Z coordinate offset of the subreflector in meters. If the focus tracking mode is on, this offset is with respect to the nominally optimum subreflector position for the current elevation. If focus tracking is off, the offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorz $\leq \infty$

Units: meters

Example of Use: proc.subreflectorz = 3.3

proc.subreflectorzrate

Description: This parameter is the Z coordinate offset velocity of the subreflector in meters per second. The subreflector begins its motion at the beginning of a scan at the position specified by the parameter 'subreflectorz'. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorzrate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectorzrate = 3.0

proc.subreflectorzstart

Description: This parameter is the start position of the Z coordinate offset of the subreflector in meters. The offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorzstart $\leq \infty$

Units: meters

Example of Use: proc.subreflectorzstart = 0.4

proc.subreflectorzstop

Description: This parameter is the stop position of the Z coordinate offset of the subreflector in meters. The offset is with respect to the optimum position at the telescope rigging elevation. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorzstop $\leq \infty$

Units: meters

Example of Use: proc.subreflectorzstop = 7.2

proc.subreflectorzstep

Description: This parameter is the Z coordinate offset step of the subreflector in the X direction in milliters. The X,Y,Z axes are defined as a right-hand coordinate system with Y away from the dish, X in the plane of away from the feed arm, and Z normal to the plane of symmetry.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectorzstep $\leq \infty$

Units: millimeters

Example of Use: proc.subreflectorzstep = 6.1

proc.subreflectoracty1

Description: This parameter is the position of subreflector actuator Y1 in meters. Increasing Y actuator length moves the subreflector toward the dish.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoracty1 $\leq \infty$

Units: meters

Example of Use: proc.subreflectoracty1 = 1.1

proc.subreflectoracty1rate

Description: This parameter is the velocity of subreflector actuator Y1 in meters per second. The postion of the actuator at the beginning of a scan is given by the value of the parameter 'subreflectoract_y1'. Increasing Y actuator length moves the subreflector toward the dish.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoracty1rate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectoracty1rate = 2.2

proc.subreflectoracty2

Description: This parameter is the position of subreflector actuator Y2 in meters. Increasing Y actuator length moves the subreflector toward the dish.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoracty2 $\leq \infty$

Units: meters

Example of Use: proc.subreflectoracty2 = 3.3

proc.subreflectoracty2rate

Description: This parameter is the velocity of subreflector actuator Y2 in meters per second. The postion of the actuator at the beginning of a scan is given by the value of the parameter 'subreflectoract_y2'. Increasing Y actuator length moves the subreflector toward the dish.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoracty2rate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectoracty2rate = 4.4

proc.subreflectoracty3

Description: This parameter is the position of subreflector actuator Y3 in meters. Increasing Y actuator length moves the subreflector toward the dish.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoracty3 $\leq \infty$

Units: meters

Example of Use: proc.subreflectoracty3 = 5.5

proc.subreflectoracty3rate

Description: This parameter is the velocity of subreflector actuator Y3 in meters per second. The postion of the actuator at the beginning of a scan is given by the value of the parameter 'subreflectoract_y3'. Increasing Y actuator length moves the subreflector toward the dish.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoracty3rate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectoracty3rate = 0.1

proc.subreflectoractx1

Description: This parameter is the position of subreflector actuator X1 in meters. Increasing X actuator length moves the subreflector away from the feed arm.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoractx1 $\leq \infty$

Units: meters

Example of Use: proc.subreflectoractx1 = 0.2

proc.subreflectoractx1rate

Description: This parameter is the velocity of subreflector actuator X1 in meters per second. The postion of the actuator at the beginning of a scan is given by the value of the parameter 'subreflectoract_x1'. Increasing X actuator length moves the subreflector away from the feed arm.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoractx1rate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectoractx1rate = 0.3

proc.subreflectoractx2

Description: This parameter is the position of subreflector actuator X2 in meters. Increasing X actuator length moves the subreflector away from the feed arm.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoractx2 $\leq \infty$

Units: meters

Example of Use: proc.subreflectoractx2 = 0.4

proc.subreflectoractx2rate

Description: This parameter is the velocity of subreflector actuator X2 in meters per second. The postion of the actuator at the beginning of a scan is given by the value of the parameter 'subreflectoract_x2'. Increasing X actuator length moves the subreflector away from the feed arm.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoractx2rate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectoractx2rate = 0.5

proc.subreflectoractz1

Description: This parameter is the position of subreflector actuator Z1 in meters. Increasing Z actuator length moves the subreflector away from the feed arm elevator.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoractz1 $\leq \infty$

Units: meters

Example of Use: proc.subreflectoractz1 = 0.6

proc.subreflectoractz1rate

Description: This parameter is the velocity of subreflector actuator Z1 in meters per second. The postion of the actuator at the beginning of a scan is given by the value of the parameter 'subreflectoract_z1'. Increasing Z actuator length moves the subreflector away from the feed arm elevator.

Values: float

Allowed Range: $-\infty \leq$ proc.subreflectoractz1rate $\leq \infty$

Units: meters per second

Example of Use: proc.subreflectoractz1rate = 0.7

proc.offsetfocus1x

Description: There is no help. yet, for offsetfocus1x

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus1x $\leq \infty$

Units: meters

Example of Use: proc.offsetfocus1x = 0.8

proc.offsetfocus1xrate

Description: There is no help. yet, for offsetfocus1xrate

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus1xrate $\leq \infty$

Units: meters per second

Example of Use: proc.offsetfocus1xrate = 0.9

proc.offsetfocus1y

Description: There is no help. yet, for offsetfocus1y

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus1y $\leq \infty$

Units: meters

Example of Use: proc.offsetfocus1y = 1.0

proc.offsetfocus1yrate

Description: There is no help. yet, for offsetfocus1yrate

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus1yrate $\leq \infty$

Units: meters per second

Example of Use: proc.offsetfocus1yrate = 1.1

proc.offsetfocus1z

Description: There is no help. yet, for offsetfocus1z

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus1z $\leq \infty$

Units: meters

Example of Use: proc.offsetfocus1z = 1.2

proc.offsetfocus1zrate

Description: There is no help. yet, for offsetfocus1zrate

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus1zrate $\leq \infty$

Units: meters per second

Example of Use: proc.offsetfocus1zrate = 1.3

proc.offsetfocus2x

Description: There is no help. yet, for offsetfocus2x

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus2x $\leq \infty$

Units: meters

Example of Use: proc.offsetfocus2x = 1.4

proc.offsetfocus2xrate

Description: There is no help. yet, for offsetfocus2xrate

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus2xrate $\leq \infty$

Units: meters per second

Example of Use: proc.offsetfocus2xrate = 1.5

proc.offsetfocus2y

Description: There is no help. yet, for offsetfocus2y

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus2y $\leq \infty$

Units: meters

Example of Use: proc.offsetfocus2y = 1.6

proc.offsetfocus2yrate

Description: There is no help. yet, for offsetfocus2yrate

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus2yrate $\leq \infty$

Units: meters per second

Example of Use: proc.offsetfocus2yrate = 1.7

proc.offsetfocus2z

Description: There is no help. yet, for offsetfocus2z

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus2z $\leq \infty$

Units: meters

Example of Use: proc.offsetfocus2z = 1.8

proc.offsetfocus2zrate

Description: There is no help. yet, for offsetfocus2zrate

Values: float

Allowed Range: $-\infty \leq$ proc.offsetfocus2zrate $\leq \infty$

Units: meters per second

Example of Use: proc.offsetfocus2zrate = 1.9

Predefined Solar System Object coordinate systems.

proc.neworigin

Description: The New Origin parameter specifies a new location for the origin (longitude = latitude = 0) of the specified Coordinate Type. If a known object is selected or orbital elements are specified, the new origin will be computed automatically to track the center of that object.

Values: Off , "User set", Orbit , Sun , Moon , Mercury , Venus , Mars , Jupiter , Saturn , Uranus , Neptune , Pluto , UserTabl

Units: N/A

Example of Use: proc.neworigin = Off

User defined coordinate systems.

proc.originlongitude

Description: The Origin Longitude is the longitude offset of the new origin in the selected Coordinate Type when the New Origin is selected as to 'User set'. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Values: float

Allowed Range: $0.0 \leq$ proc.originlongitude ≤ 360.0

Units: degrees

Example of Use: proc.originlongitude = 23.2

proc.originlongvel

Description: The Origin Longitude Velocity is the rate of change of the longitude offset of the new origin in the selected Coordinate Type when the New Origin is selected as to 'User set'. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Values: float

Units: degrees per second

Example of Use: proc.originlongvel = 0.1

proc.originlatitude

Description: The Origin Latitude is the latitude offset of the new origin in the selected Coordinate Type when the New Origin is selected as to 'User set'. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Values: float

Allowed Range: -90.0 $\leq$ proc.originlatitude $\leq$ +90.0

Units: degrees

Example of Use: proc.originlatitude = 67.3

proc.originlatvel

Description: The Origin Latitude Velocity is the rate of change of the latitude offset of the new origin in the selected Coordinate Type when the New Origin is selected as to 'User set'. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Values: float

Units: degrees per second

Example of Use: proc.originlatvel = 0.01

proc.originrotation

Description: The Origin Rotation is the rotation the new origin's equator with respect to the equator in selected Coordinate Type when the New Origin is selected as 'User set'. Positive rotation is counterclockwise as the observer sees it on the sky. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Values: float

Allowed Range: $0.0 \leq$ proc.originrotation ≤ 360.0

Units: degrees

Example of Use: proc.originrotation = 136.4

proc.originrotvel

Description: The Origin Rotation Velocity is the rate of rotation of the new origin's equator with respect to the equator in selected Coordinate Type when the New Origin is selected as 'User set'. Positive rotation is counterclockwise as the observer sees it on the sky. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Values: float

Units: degrees per second

Example of Use: proc.originrotvel = 0.0234

proc.originstartutc

Description: The Origin Start Utc is the UTC for which the new origin longitude and latitude are valid when any of the rates, 'originlongvel', 'originlatvel', or 'originlatvel', is not zero. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Units: HH:MM:SS.S

Example of Use: proc.originstartutc = "10:10:10"

proc.originstartlst

Description: The Origin Start Lst is the LST for which the new origin longitude and latitude are valid when any of the rates, 'originlongvel', 'originlatvel', or 'originlatvel', is not zero. The Origin Start LST is immediately converted to Origin Start UTC, which is the primary time system. The New Origin lets you specify the location of a temporay new origin for your coordinate grid so that the telescope beam position can specified with respect to the new origin. This origin may move at a specified rate in longitude and latitude, and the new equator may be rotated with respect to the parent Coordinate Type equator. The new origin is fully specified by the parameters 'originlongitude', 'originlongvel', 'originlatitude', 'originlatvel', 'originrotation', 'originrotvel', and 'originstartutc' or 'originstartlst'. The coordinate value units are degrees and degrees per second.

Units: HH:MM:SS.S

Example of Use: proc.originstartlst = "11:11:11"

proc.orbitrefframe

Description: The Orbit Reference Frame defines the inertial frame and spherical coordinate system in which the orbit is defined by the orbital parameters. The choices are EARTH_B1950, EARTH_J2000, ECLIPTIC_B1950, and ECLIPTIC_J2000. The first two are use equatorial coordinates for earth-orbiting satellites, and the others are used for solar system objects in ecliptic coordinates using either the B1950 or J2000 vernal equinox.

Values: " Earth B1950 ", " Earth J2000 ", "Sun/Ecliptic B1950", "Sun/Ecliptic J2000"

Units: N/A

Example of Use: proc.orbitrefframe = "Earth J2000"

proc.orbitepoch

Description: The Orbit Epoch is the fractional Modified Julian Date (MJD = JD - 2400000.5) for which the orbital elements are valid. In particular, if the Mean Anomaly is specified instead of Pericenter Epoch to define the position of the object in its orbit, this Mean Anomaly is at the the time of the Orbit Epoch. In that case the orbit epoch must be entered with enough precision to define the object's position accurately.

Values: float

Units: MJD fractional days

Example of Use: proc.orbitepoch = 52893.456

proc.pericenterepoch

Description: The Pericenter Epoch is the time, in fractional Modified Julian Date (MJD = JD - 2400000.5) when the orbiting object passes pericenter. The position of the object in its orbit can alternatively be specified by its Mean Anomaly, which is its mean position, in degrees, at the time given by the Orbit Epoch. Calculations of the object's position as a function of time will use the most recently specified of these two parameters.

Values: float

Units: MJD fractional days

Example of Use: proc.pericenterepoch = 51345.67

proc.meananomaly

Description: The Mean Anomaly is the mean position of the object at the time specified by the Orbit Epoch. The mean position is the angular distance, in degrees, from pericenter of the object if it had a uniform angular velocity throughout its orbit. This is given by 360 multiplied by the time since pericenter passage divided by the orbit period. The position of the object in its orbit can alternatively be specified by its Pericenter Epoch. Calculations of the object's position as a function of time will use the most recently specified of these two parameters.

Values: float

Allowed Range: $0.0 \leq$ proc.meananomaly ≤ 360.0

Units: degrees

Example of Use: proc.meananomaly = 15.9

proc.semimajoraxis

Description: The Semimajor Axis of the object's orbit is specified in kilometers. The size of the orbit may alternatively be specified by its Orbit Period or its Mean Daily Motion. These assume either the mass of the sun or mass of the earth, as implied from the selected Orbit Reference Frame, to derive the Semimajor Axis. The most recently specified of the three parameters will be used to compute the values of the other two.

Values: float

Allowed Range: $0.0 \leq$ proc.semimajoraxis $\leq \infty$

Units: kilometers

Example of Use: proc.semimajoraxis = 259876.3

proc.orbitperiod

Description: The Orbital Period is specified in days. Alternatively, the object's Mean Daily Motion may be specified, where the Orbital Period = (360 / Mean Daily Motion). Specifying either of these two parameters will cause the Semimajor Axis to be computed with the assumption of the mass of the sun or mass of the earth, as implied from the selected Orbit Reference Frame. The most recently specified of the three parameters, Semimajor Axis, Orbital Period, or Mean Daily Motion, will be used to compute the values of the other two.

Values: float

Allowed Range: $0.0 \leq$ proc.orbitperiod $\leq \infty$

Units: fractional days

Example of Use: proc.orbitperiod = 365.249

proc.meandailymotion

Description: The Mean Daily Motion is the average angular velocity, in degrees per day, of the object in its orbit as computed from (360 degrees / Orbit Period in days). Alternatively, the object's Orbital Period may be specified. Specifying either of these two parameters will cause the Semimajor Axis to be computed with the assumption of the mass of the sun or mass of the earth, as implied from the selected Orbit Reference Frame. The most recently specified of the three parameters, Semimajor Axis, Orbital Period, or Mean Daily Motion, will be used to compute the values of the other two.

Values: float

Units: degrees per day

Example of Use: proc.meandailymotion = 1.0

proc.eccentricity

Description: Eccentricity of the orbit is pretty well self-evident. Alternatively, the Pericenter Distance may be specified. The most recently specified of the two parameters will be used to compute the value of the other using the current value of the Semimajor Axis.

Values: float

Allowed Range: $0.0 \leq$ proc.eccentricity $\leq \infty$

Units: N/A

Example of Use: proc.eccentricity = 0.95

proc.pericenterdistance

Description: The Pericenter Distance is the distance, in kilometers, of the orbiting object from the central mass at closest approach. Alternatively, the orbital Eccentricity may be specified. The most recently specified of the two parameters will be used to compute the value of the other using the current value of the Semimajor Axis.

Values: float

Allowed Range: $0.0 \leq$ proc.pericenterdistance $\leq \infty$

Units: kilometers

Example of Use: proc.pericenterdistance = 6345.65

proc.pericenterargument

Description: The Pericenter Argument is the angular distance, in degrees, from the ascending node to the pericenter in the plane of the orbit.

Values: float

Allowed Range: $0.0 \leq$ proc.pericenterargument ≤ 360.0

Units: degrees

Example of Use: proc.pericenterargument = 23.2

proc.longascendingnode

Description: The Longitude of the Ascending Node is the angular distance, in degrees, from the vernal equinox to the point where the orbiting object passes through the reference plane (equator or ecliptic) from south to north.

Values: float

Allowed Range: 0.0 ≤ proc.longascendingnode ≤ 360.0

Units: degrees

Example of Use: proc.longascendingnode = 159.4

proc.orbitinclination

Description: The Orbit Inclination is angle, in degrees, between the orbital plane and the reference plane (equator or ecliptic). Its value is between 0 and 90 degrees.

Values: float

Allowed Range: 0.0 ≤ proc.orbitinclination ≤ 90.0

Units: degrees

Example of Use: proc.orbitinclination = 1.9

4.2 Scan Coordinator

- sc.nextscannumber
- sc.sourcename
- sc.projid
- sc.scanid
- sc.scanlength
- sc.starttime
- sc.stoptime
- sc.observername
- sc.logfile
- sc.obstype
- sc.switchingsignalsmaster
- sc.calstate
- sc.sigrefstate
- sc.blankingtime
- sc.numberofphases
- sc.phasestart
- sc.switchmode

- sc.subsystemselect
- sc.receiver

sc.nextscannumber

Description: The Next Scan Number allows you to set the scan number to a new value for the next scan executed.

Values: integer

Allowed Range: $0 \leq$ sc.nextscannumber $\leq \infty$

Units:

Example of Use: sc.nextscannumber = 1

sc.sourcename

Description: Any Source Name less than 32 characters. In pulsar observing this name is used to fetch the current pulsar timing parameters and must be in a standard 'hhmmsdd' format, e.g., 0329+54. In other observing modes the Source Name is only an identifier label.

Values: any string ¡ 32 Characters

Units: N/A

Example of Use: sc.sourcename = "3C 218"

sc.projid

Description: The Project ID is the number assigned to your program on the telescope schedule, e.g., B345. This string is used as a directory name for your data.

Values: any string ¡ 32 Characters

Units: N/A

Example of Use: sc.projid = "AGBT01A_999_01"

sc.scanid

Description: The Scan I.D. is a supplement to the Source Name for labeling the data. It has no effect on the data taking process. The string must be less than 32 characters.

Values: any string ¡ 32 Characters

Units: N/A

Example of Use: sc.scanid = "Tracking 3C 318 for IPS"

The Scan Length is the duration of the scan in seconds. Any data integrations completed after the end of a scan will normally be discarded. Hence, the Scan Length is typically an integer number of integration times plus a second or two. 0∞ sc.scanlength = 3000

sc.starttime

Description: The Start Time may be explicitly specified in UTC or LST, or it may simply be as-soon-as-possible. Select the start or stop mode and the type of time with the menu buttons. In a.s.a.p. mode the time displayed is what was actually used for the last scan. Normally the date, and hence the MJD, is implied from the current date. Times are assumed to be in the interval between 1/2 hour before and 23 1/2 hours after the current time. The date may be explicitly set in UTC mode only by unlocking the Date/MJD fields with the glish command "set_mjd_auto(F)". The date may be entered in either mm/dd/yy or MJD format. Use "set_mjd_auto(T)" to relock it. LST times ignore the date settings and use the current date. In Stop Time mode the scan start as soon possible.

Units: HH:MM:SS.S

Example of Use: sc.starttime = "10:09:08"

sc.stoptime

Description: The Stop Time may be explicitly specified in UTC or LST, or it may simply be as-soon-as-possible. Select the start or stop mode and the type of time with the menu buttons. In a.s.a.p. mode the time displayed is what was actually used for the last scan. Normally the date, and hence the MJD, is implied from the current date. Times are assumed to be in the interval between 1/2 hour before and 23 1/2 hours after the current time. The date may be explicitly set in UTC mode only by unlocking the Date/MJD fields with the glish command "set_mjd_auto(F)". The date may be entered in either mm/dd/yy or MJD format. Use "set_mjd_auto(T)" to relock it. LST times ignore the date settings and use the current date. In Stop Time mode the scan start as soon possible.

Units: HH:MM:SS.S

Example of Use: sc.stoptime = "23:59:59"

sc.observername

Description: The Observer's Name can be any string. This is recorded with the data.

Values: any string

Units: N/A

Example of Use: sc.observername = "Alfred Nobel"

sc.logfile

Description: Glish Log File is the name of the file that records all glish language commands generated by 'GBT Observe', either from GUI selections and entries, from the command line, or from observing table execution

Values: filename

Units: N/A

Example of Use: sc.logfile = "/users/anobel/obs/go.log"

sc.obstype

Description: Observing Type determines the parameters requested on the left side of the Main Screen and selects the primary backend to be used. It also sets a lot of default values for various devices. Make this selection before setting parameters for specific devices.

Values: Continuum, ”Spectral Line SPM”, ”Spectral Line SP”, ”Pulsar Timing SP”, ”Pulsar Dedispersion”, ”Pulsar Search BCPM”, ”Pulsar Voltage Sampling BCPM”, ”Pulsar Timing BCPM”, ”Pulsar Monitoring BCPM”, ”Undefined”

Units: N/A

Example of Use: sc.obstype = ”Spectral Line SPM”

sc.switchingsignalsmaster

Description: The Switching Signals Master selects which backend provides the switching signals to all of the backends.

Values: SpectralProcessor, DCR, Spectrometer, VLBADAR, GBPP

Units: N/A

Example of Use: sc.switchingsignalsmaster = Spectrometer

sc.calstate

Description: Each Cal toggle button specifies the state of the receiver calibration signal in the button’s switching phase. The number of phases depends on the selected Switching Mode or on the number of phases selected by the user in the ”User Defined” mode. In all but the ”User Defined” mode the Cal states are predetermined by the selected mode.

Values: Off, On

Max Size: 64

Units: N/A

Example of Use: sc.calstate[1] = Off
sc.calstate[2] = On

sc.sigrefstate

Description: Each SigRef toggle button specifies the state of the receiver frequency/load/beam switch signal in the button’s switching phase. The number of phases depends on the selected Switching Mode or on the number of phases selected by the user in the ”User Defined” mode. In all but the ”User Defined” mode the SigRef states are predetermined by the selected mode.

Values: Sig, Ref

Max Size: 64

Units: N/A

Example of Use: sc.sigrefstate[1] = Sig
sc.sigrefstate[2] = Ref

sc.blankingtime

Description: Banking Time is the time in seconds at the beginning of each switch phase when data integration is inhibited. The Blanking Time may be set to a different value for each phase through the glish command line, but one value is usually sufficient for every phase, and that value is specified here. The resolution in 100nS.

Values: float

Max Size: 64

Units: seconds

Example of Use: sc.blankingtime = 0.003

sc.numberofphases

Description: The Number of Phases specifies how many phases are in the switching cycle. This number is predetermined by the selected Switching Mode for all but the ”User Defined” mode. In that mode the number may be between 1 and 10.

Values: integer

Allowed Range: $0 \leq$ sc.numberofphases $\leq \infty$

Units: N/A

Example of Use: sc.numberofphases = 2

sc.phasestart

Description: Each Phase Start entry field specifies the beginning of this phase as a fraction of the total switch cycle. The first start time must be zero, they must increase monotonically, and the last phase start time must be less than one. The effective integration time for a phase in one switching cycle is the product of the Switch Period and the difference between that phase's and the next phase's start times minus the Blanking Time. The number of phases depends on the selected Switching Mode or on the number of phases selected by the user in the ”User Defined” mode. In all but the ”User Defined” mode the phase Start times are predetermined by the selected mode.

Values: float

Max Size: 64

Units: seconds

Example of Use: sc.phasestart[1,2] = [0,0.5]

sc.switchmode

Description: The Switching Mode is a menu of predefined switching modes plus a user-defined mode. The switching parameters may be displayed with the Switching Setup button. The Number of Phases, SigRef and Cal states, phase Start times, and Advance Sig selections are all set when one of the predefined modes is selected. In the ”User Defined” mode all of these parameters are available for user input.

Values: ”Total Power”, ”Total Power, No Cal”, ”Total Power, Spec Proc”, ”Freq Switch, 01”, ”Freq Switch, 12”, ”Freq Switch, 0102”, ”Beam Switch”, ”Pol. Switch”, ”User Defined”

Units: N/A

Example of Use: sc.switchmode = ”Freq Switch, 01”

sc.subsystemselect

Description: The Subsystem Select parameters determines which of the GBT subsystems are active to participate in scan sequence. Normally, this parameter is set automatically as a result of selection of the receiver and observing type. The value for this parameter is a 23-element, boolean array with the elements corresponding, respectively, to the ”Antenna”, ”LO1”, ”0.3-0.9 GHz Rcvr”, ”1.2-1.7 GHz Rcvr”, ”1.7-2.6 GHz Rcvr”, ”3.9-5.8 GHz Rcvr”, ”8-10 GHz Rcvr”, ”12-15 GHz Rcvr”, ”18-26 GHz Rcvr”, ”40-52 GHz Rcvr”, ”I.F. Rack”, ”Converter Rack”, ”Analog Filt Rack”, ”Sw Signal Selector”, ”Spectral Proc.”, ”DCR”, ”Holography”, ”Spectrometer”, ”BCPM”, ”Archivist”, ”Measurements”, ”Active Surface” and ”I.F. Manager”.

Values: T, F

Max Size: 23

Units: N/A

Example of Use: sc.subsystemselect[1] = T

sc.receiver

Description: The Receiver parameter, in conjunction with the Observing Type, determines a lot of default settings the setup up the GBT system to use the selected receiver. Make this selection before setting parameters for specific devices.

Values: ”NoiseSource”, ”0.290 - 0.395”, ”0.385 - 0.520”, ”0.510 - 0.690”, ”0.680 - 0.920”, ”0.910 - 1.230”, ”1.15 - 1.73”, ”1.73 - 2.60”, ”3.95 - 5.85”, ”8.00 - 10.1”, ”12.0 - 15.4”, ”18.0 - 22.4”, ”22.0 - 26.5”, ”40.0 - 50.0”

Units: N/A

Example of Use: sc.receiver = ”1.15 - 1.73”

4.3 Antenna

- ant.coordinatemode
- ant.offsetcoordinatemode
- ant.primaryunits
- ant.epoch
- ant.cosminormode
- ant.azimuthwrap
- ant.elevationwrap
- ant.trackingbeam
- ant.beam

ant.coordinatemode

Description: The Coordinate Mode defines the coordinate system in which the direction of the telescope beam is specified. This coordinate system applies to the Primary Segment and Primary Offset parameters and the coordinate system into which the User Transform coordinates are rotated. The available Coodinate Modes are J2000, B1950, RaDecofDate, ApparentRaDec, Galactic, HaDec, and AzEl.

Values: J2000 , B1950 , RaDecOfDate , ApparentRaDec, Galactic , HaDec , AzEl , Encoder

Units: N/A

Example of Use: ant.coordinatemode = ”B1950”

ant.offsetcoordinatemode

Description: The Offset Coordinate Mode defines the coordinate system in which the direction of the telescope motion is specified. This coordinate system applies to the Primary Offset parameters. The available Offset Coodinate Modes are J2000, B1950, RaDecofDate, ApparentRaDec, Galactic, HaDec, and AzEl.

Values: J2000 , B1950 , RaDecOfDate , ApparentRaDec, Galactic , HaDec , AzEl , Encoder

Units: N/A

Example of Use: ant.offsetcoordinatemode = ”Galactic”

ant.primaryunits

Description: The Primary Units parameter specifies the units used by the Primary Segments and Primary Offsets. The choices are Hrs/Deg, Deg/Deg, and Rad/Rad. Hrs/Deg normally only applies to RA/Dec and HA/Dec but can be applied to other coordinates.

Values: Deg/Deg, Hrs/Deg, Rad/Rad

Units: N/A

Example of Use: ant.primaryunits = ”Deg/Deg”

ant.epoch

Description: This parameter is the epoch of the coordinateMode. Active only when coordinateMode is set to RaDecOfDate. Otherwise this value provides feedback.

Values: float

Units: Fraction year

Example of Use: ant.epoch = 2003.76

ant.cosminormode

Description: The Cosine Minor parameter controls whether position offsets in the major coordinate (RA, Azimuth, or Longitude) are multiplied by the secant of the minor coordinate (Dec, Elevation, or Latitude). The choices are On and Off.

Values: On , Off

Units: N/A

Example of Use: ant.cosminormode = On

ant.azimuthwrap

Description: The Azimuth Wrap parameter controls which of the two possible azimuth cable wrap positions are used for a given telescope azimuth in the ranges where there are two possibilities. The default is 'Auto' which lets the telescope control software determine the best choice. The value 'CCW' keeps the telescope azimuth in the range -90 to +270 degrees, and the value 'CW' keeps it in the range +90 to +450 degrees. Zero degrees is north and +90 degrees is east.

Values: Auto, CCW , CW

Units: N/A

Example of Use: ant.azimuthwrap = CCW

ant.elevationwrap

Description: The Elevation Wrap parameter controls which of the two possible telescop elevation positions are used near the zenith. The value 'Auto' keeps the elevation less than 90 degrees. The value 'Over the Top' uses the range 90 to 95 degrees, which requires an azimuth position 180 degrees from what it would be in the 'Auto' position for a given place on the sky.

Values: Auto , "Over Top"

Units: N/A

Example of Use: ant.elevationwrap = "Auto"

ant.trackingbeam

Description: The tracking_beam parameter determines which feed of the receiver is to track the source (i.e. beam on-source)

Values: "1", "2", "3", "4", "M12", "M34", "C"

Units: N/A

Example of Use: ant.trackingbeam = "1"

ant.beam

Description: The beam parameter determines which feed of the receiver is to track the source (i.e. be on-source)

Values: "1", "2", "3", "4", "M12", "M34", "C"

Units: N/A

Example of Use: ant.beam = "2"

4.4 LO System

4.4.1 LO1

- lo1.lo_config
- lo1.def_vel chosen then the src_vel keyword will specify the unit-less value of the
- lo1.src_vel

- lo1.numfswoffsets
- lo1.ref_freq_1 The reference frequency offset 1 in MHz. This keyword is only used when
- lo1.ref_freq_2 The reference frequency offset 2 in MHz. This keyword is only used when
- lo1.ref_frame
- lo1.rest_freq
- lo1.tolerance
- lo1.testtone_freq
- lo1.if_center_freq
- lo1.sideband sideband_b keyword values to the current value of sideband.
- lo1.power_level
- lo1.auto_power_level
- lo1.testtone_power_level
- lo1.use_offsets
- lo1.counter_band
- lo1.counter_resolution
- lo1.s1
- lo1.s2
- lo1.s3
- lo1.s4
- lo1.s5
- lo1.s6
- lo1.s7
- lo1.s8
- lo1.s9
- lo1.s10
- lo1.s11
- lo1.s12
- lo1.s13
- lo1.s14
- lo1.s15

lo1.lo_config

Description: Defines the configuration of LO1A and LO1B. Either LO1A or LO1B can be used as a tracking LO, with the other unused or operating as a test tone generator.

Values: "TrackA_TToneB","TrackB_TToneA", "TrackA_BNotUsed","TrackB_ANotUsed"

Units: N/A

Example of Use: lo1.lo_config = "TrackA_BNotUsed"

lo1.def_vel

Description: The velocity definition which specifies how the source velocity is translated into frequency for Doppler tracking. Possible choices are 'RELATIVISTIC', 'OPTICAL', 'RADIO', or 'Z'. If redshift (z) is chosen then the src_vel keyword will specify the unit-less value of the redshift.

N.B. **The 'Z' option is not yet implemented in YGOR.**

Values: RELATIVISTIC, OPTICAL, RADIO

Units: N/A

Example of Use: lo1.def_vel = "RADIO"

lo1.src_vel

Description: The velocity, in units of km/s, which determines the sky frequency of the spectrometer passband. The Doppler correction equations are used to convert this velocity to frequency using the rest frame specified by the velocity definition and reference frame. It is assumed that this velocity is constant (no acceleration or jerk terms) and that it is valid for all times (i.e. epoch=0).

Values: float

Units: km/s or unit-less if def_vel = Z

Example of Use: lo1.src_vel = 100.03

lo1.numfswoffsets

Description: The number of frequency switching offsets determines the size of the fsw_offsets array. The number of frequency switching offsets must be larger than zero and less than 5. Changing the number of frequency offsets will change the fsw_offsets array and possibly the values of rest_freq_1 and rest_freq_2.

Values: integer

Allowed Range: $0 \leq$ lo1.numfswoffsets ≤ 5

Units: N/A

Example of Use: lo1.num_fsw_offsets = 2

lo1.ref_freq_1

Description: The reference frequency offset 1 in MHz. This keyword is only used when the switching mode is set to one of the frequency switching options.

Values: float

Units: MHz

Example of Use: lo1.freq_freq_1 = 2.5

lo1.ref_freq_2

Description: The reference frequency offset 2 in MHz. This keyword is only used when the switching mode is set to one of the frequency switching options.

Values: float

Units: MHz

Example of Use: lo1.ref_freq_1 = -2.5

lo1.ref_frame

Description: Inertial reference frame for Doppler tracking. The source velocity is expressed in terms of the selected rest frame. Possible values are 'Local', 'Barycenteric', 'Heliocentric', 'LSR', 'LSRD', 'Galactocentric', 'Localgroup', and 'CMB'.

Local local topocentric rest frame of the telescope.

Barycentric the center of mass of the solar system.

Heliocentric the center of the Sun.

LSR or LSRK the kinematic local standard of rest which is a point in the vicinity of the Sun which has the motion of 20 km/s toward RA=18:00:00.0, Dec=30:00:00 (1900).

LSRD the dynamical local standard of rest which is a point in the vicinity of the Sun in a circular orbit around the Galactic Center. This peculiar motion is 16.6 km/s toward RA=17:49:58.7, Dec=28:07:04 (J2000).

Galactocentric the Galactic Center and is referenced from the dynamical LSR rest frame assuming the Sun is moving 220 km/s toward Ra=21:12:01.1, Dec=48:19:47 (J2000).

Localgroup defined as the standard of rest with respect to the Local Group of Galaxies.

CMB defined as the standard of rest with respect to the Cosmic Microwave Background.

N.B. The Localgroup and CMB reference frames are not yet implemented in the YGOR LO1 manager.

N.B. At some observatories heliocentric really means barycentric. In general there are two different LSR frames: kinematic and dynamical. Since LSR generally corresponds to the kinematic local standard of rest LSR=LSRK. LSRK is derived from "standard solar motion" while LSRD is derived from "basic solar motion."

Values: Local, Barycentric, Heliocentric, LSR, LSRD, Galactocentric, Localgroup, Cmd

Units: N/A

Example of Use: lo1.ref_frame = "LSR"

lo1.rest_freq

Description: The frequency with respect to the astronomical object in MHz (i.e., no Doppler correction). For example, observations of the 21cm line of HI would set the rest frequency to 1420.4058 MHz. For continuum observations it is the sky center frequency of the final passband when Doppler tracking is not required. In continuum observations, the following should also be set:
ref_frame = TOPOCENTER, vel_def = RADIO, and src_vel = 0.

Values: float

Units: MHz

Example of Use: lo1.rest_freq = 1420.4058

lo1.tolerance

Description: The desired frequency tolerance in Hz for Doppler updates. The minimum value is 1 Hz.

Values: float

Units: Hz

Example of Use: lo1.tolerance = 10.0

lo1.testtone_freq

Description: The frequency of the testtone signal in MHz.

Values: float

Units: MHz

Example of Use: lo1.testtone_freq = 1245.67

lo1.if_center_freq

Description: The desired IF center frequency after the first mixer (LO1) in MHz. Effectively the sky frequency and the IF center frequency are used to determine the LO1 frequency. This value depends on the front end receiver chosen and will be set by default to the following values when the effective receiver is selected: 0.290 - 0.395 GHz (1080 MHz), 0.385 - 0.520 GHz (1080 MHz), 0.510 - 0.690 GHz (1080 MHz), 0.680 - 0.920 GHz (1080 MHz), 0.910 - 1.230 GHz (1080 MHz), 1.15 - 1.73 GHz (3000 MHz), 1.73 - 2.60 GHz (6000 MHz), 3.95 - 5.85 GHz (3000 MHz), 8.0 - 10.1 GHz (3000 MHz), 12.0 - 15.4 GHz (3000 MHz), 18.0 - 22.4 GHz (6000 MHz), 22.0 - 26.5 GHz (6000 MHz), 40.0 - 50.0 GHz (6000 MHz). These default values may be overridden, however, by using the 'if_center_frequency' parameter.

Values: float

Units: MHz

Example of Use: lo1.if_center_freq = 3000.0

lo1.sideband

Description: The LO1 synthesizer sideband. Possible values are 'upper' or 'lower'. The sideband will be set by default when the effective receiver is selected using the following values: 0.290 - 0.395 GHz (lower), 0.385 - 0.520 GHz (lower), 0.510 - 0.690 GHz (lower), 0.680 - 0.920 GHz (lower), 0.910 - 1.230 GHz (lower), 1.15 - 1.73 GHz (lower), 1.73 - 2.60 GHz (lower), 3.95 - 5.85 GHz (lower), 8.0 - 10.1 GHz (lower), 12.0 - 15.4 GHz (upper), 18.0 - 22.4 GHz (upper), 22.0 - 26.5 GHz (upper), 40.0 - 50.0 GHz (upper). The sideband parameter automatically sets both the sideband_a and sideband_b keyword values to the current value of sideband.

Values: upper, lower

Units: N/A

Example of Use: lo1.sideband = "upper"

lo1.power_level

Description: The LO1 synthesizer output power level in dBm. This is typically set automatically using the auto_power_level parameter when a receiver is selected. Possible values range from -20 to +13.5dBm.

Values: float

Allowed Range: -20 ≤ lo1.power_level ≤ +13.5

Units: dBm

Example of Use: lo1.power_level = 2.5

lo1.auto_power_level

Description: Enables LO1 automatic power level setting. If set to 'ON' the LO1 synthesizer output power level will be set automatically depending on the selected receiver. Otherwise if 'OFF' the parameter 'power_level' is used to set the output power level. The automatic power level is optimized from previous experience for each receiver.

Values: T, F

Units: N/A

Example of Use: lo1.auto_power_level = "ON"

lo1.testtone_power_level

Description: The testtone synthesizer output power level in dBm. Possible values range from -20 to +13.5dBm.

Values: float

Allowed Range: -20 ≤ lo1.testtone_power_level ≤ 13.5

Units: dBm

Example of Use: lo1.testtone_power_level = 4.5

lo1.use_offsets

Description: A boolean to determine if the offset positions are used to when calculating frequency information (e.g., Doppler tracking). The default setting is 'ON'.

Values: T, F

Units: N/A

Example of Use: lo1.use_offsets = "OFF"

lo1.counter_band

Description: This parameter determines which band the frequency counter will use.

Values: "band 1", "band 2", "band 3"

Units: N/A

Example of Use: lo1.counter_band = "band 1"

lo1.counter_resolution

Description: Resolution of the counter and by extension the sample rate.

Values: "1 Hz", "10 Hz", "100 Hz", "1 kHz", "10 kHz"

Units: N/A

Example of Use: lo1.counter_resolution = "10 Hz"

lo1.s1

Description: A copy of the LO1A and LO1B synthesizer frequencies are available to be routed to the LO Counter or to the Test Tone outputs of the LO1 rack. One input signal (LO1A or LO1B generated) into the S1 switch is routed on to be available for the LO Counter or the Test Tone outputs (via the S3 switch) while the other input signal into the S1 switch is terminated to ground. When the S1 switch is in the "thru" position the LO1B generated signal is terminated to ground while the LO1A generated signal is routed on to the S3 switch. When the S1 switch is in the "cross" position the LO1A generated signal is terminated to ground while the LO1A generated signal is routed on to the S3 switch. The signal routed on to the S3 switch will be referred to as the "LO monitor signal".

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: "thru", "cross"

Units: N/A

Example of Use: lo1.s1 = "thru"

lo1.s2

Description: A copy of the LO1A and LO1B synthesizer frequencies are available to be routed to the receiver LO inputs. One input signal (LO1A or LO1B generated) into the S2 switch is routed on to be available to the LO receiver output ports 1-18 of the LO1 rack (via the S4 and S5, S6 & S7 switches) while the other input signal into the S2 switch is routed on to be available to the LO receiver output ports 19-24 of the LO1 rack (via the S8 switch). When the S2 switch is in the "thru" position the LO1A generated signal is routed on to the S4 switch while the LO1B generated signal is routed on to the S8 switch. When the S2 switch is in the "cross" position the LO1B generated signal is routed on to the S4 switch while the LO1A generated signal is routed on to the S8 switch. The signal routed on to the S4 switch will be referred to as the "LO signal".

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: "thru", "cross"

Units: N/A

Example of Use: lo1.s2 = "thru"

lo1.s3

Description: Either of the LO1A or the LO1B generated LO signals (the "LO monitor signal") can be available to be routed to the LO Counter or the Test Tone outputs of the LO1 rack. The S3 switch determines where the LO monitor signal is routed. The S3 switch can be in 5 different positions labeled numerically by an integer in the range 0 to 4. When the S3 switch is set to 0 (zero) there is no power on the LO monitor signal and the S3 switch is in a "undefined" state. When the S3 switch is set to 1 the LO monitor signal is sent to the High Resolution LO Counter. When the S3 switch is set to 2 the LO monitor signal is sent to the Low Resolution LO Counter. When the S3 switch is set to 3 the LO monitor signal is made available to the Test Tone outputs by routing the LO monitor signal to the S11 switch. When the S3 switch is set to 4 the LO monitor signal is terminated to ground.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $0 \leq$ lo1.s3 ≤ 4

Units: N/A

Example of Use: lo1.s3 = 2

lo1.s4

Description: Either of the LO1A or the LO1B generated LO signals (the "LO signal") can be available to be routed to the LO receiver output ports 1-24 of the LO1 rack. The S4 switch determines where the LO signal is routed. The S4 switch can be in 5 different positions labeled numerically by an integer in the range 0 to 4. When the S4 switch is set to 0 (zero) there is no power on the LO signal and the S4 switch is in a "undefined" state. When the S4 switch is set to 1 the LO signal is sent to the LO receiver output ports 1-6 of the LO1 rack via the S5 switch. When the S4 switch is set to 2 the LO signal is sent to the LO receiver output ports 7-12 of the LO1 rack via the S6 switch. When the S4 switch is set to 3 the LO signal is sent to the LO receiver output ports 13-18 of the LO1 rack via the S7 switch. When the S4 switch is set to 1 the LO signal is terminated to ground.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s4 ≤ 4

Units: N/A

Example of Use: lo1.s4 = 2

lo1.s5

Description: The S5 switch routes the LO signal output from the S4 switch to one of the LO receiver output ports (1-6) of the LO1 rack. The S5 switch can be in 7 different positions labeled numerically by an integer in the range 0 to 6. When the S5 switch is set to 0 (zero) there is no power on the LO signal and the S5 switch is in a "undefined" state. When the S5 switch is set to 1 the LO signal is sent to the LO receiver output port 1 of the LO1 rack. When the S5 switch is set to 2 the LO signal is sent to the LO receiver output port 2 of the LO1 rack. When the S5 switch is set to 3 the LO signal is sent to the LO receiver output port 3 of the LO1 rack. When the S5 switch is set to 4 the LO signal is sent to the LO receiver output port 4 of the LO1 rack. When the S5 switch is set to 5 the LO signal is sent to the LO receiver output port 5 of the LO1 rack. When the S5 switch is set to 6 the LO signal is sent to the LO receiver output port 6 of the LO1 rack.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s5 ≤ 6

Units: N/A

Example of Use: lo1.s5 = 2

lo1.s6

Description: The S6 switch routes the LO signal output from the S4 switch to one of the LO receiver output ports (7-12) of the LO1 rack. The S6 switch can be in 7 different positions labeled numerically by an integer in the range 0 to 6. When the S6 switch is set to 0 (zero) there is no power on the LO signal and the S6 switch is in a "undefined" state. When the S6 switch is set to 1 the LO signal is sent to the LO receiver output port 7 of the LO1 rack. When the S6 switch is set to 2 the LO signal is sent to the LO receiver output port 8 of the LO1 rack. When the S6 switch is set to 3 the LO signal is sent to the LO receiver output port 9 of the LO1 rack. When the S6 switch is set to 4 the LO signal is sent to the LO receiver output port 10 of the LO1 rack. When the S6 switch is set to 5 the LO signal is sent to the LO receiver output port 11 of the LO1 rack. When the S6 switch is set to 6 the LO signal is sent to the LO receiver output port 12 of the LO1 rack.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s6 ≤ 6

Units: N/A

Example of Use: lo1.s6 = 2

lo1.s7

Description: The S7 switch routes the LO signal output from the S4 switch to one of the LO receiver output ports (13-18) of the LO1 rack. The S7 switch can be in 7 different positions labeled numerically by an integer in the range 0 to 6. When the S7 switch is set to 0 (zero) there is no power on the LO signal and the S7 switch is in a "undefined" state. When the S7 switch is set to 1 the LO signal is sent to the LO receiver output port 13 of the LO1 rack. When the S7 switch is set to 2 the LO signal is sent to the LO receiver output port 14 of the LO1 rack. When the S7 switch is set to 3 the LO signal is sent to the LO receiver output port 15 of the LO1 rack. When the S7 switch is set to 4 the LO signal is sent to the LO receiver output port 16 of the LO1 rack. When the S7 switch is set to 5 the LO signal is sent to the LO receiver output port 17 of the LO1 rack. When the S7 switch is set to 6 the LO signal is sent to the LO receiver output port 18 of the LO1 rack.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s7 ≤ 6

Units: N/A

Example of Use: lo1.s7 = 2

lo1.s8

Description: Either of the LO1A or the LO1B generated LO signals (the "LO signal") can be available to be routed to the LO receiver output ports 19-24 of the LO1 rack. The S8 switch determines where the LO signal output from the S2 switch is routed. The S8 switch can be in 7 different positions labeled numerically by an integer in the range 0 to 6. When the S8 switch is set to 0 (zero) there is no power on the LO signal and the S8 switch is in a "undefined" state. When the S8 switch is set to 1 the LO signal is sent to the LO receiver output port 19 of the LO1 rack. When the S8 switch is set to 2 the LO signal is sent to the LO receiver output port 20 of the LO1 rack. When the S8 switch is set to 3 the LO signal is sent to the LO receiver output port 21 of the LO1 rack. When the S8 switch is set to 4 the LO signal is sent to the LO receiver output port 22 of the LO1 rack. When the S8 switch is set to 5 the LO signal is sent to the LO receiver output port 23 of the LO1 rack. When the S8 switch is set to 6 the LO signal is sent to the LO receiver output port 24 of the LO1 rack.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s8 ≤ 6

Units: N/A

Example of Use: lo1.s8 = 2

lo1.s9

Description: The S9 switch determines which synthesizer is used to generate the LO1A LO signal. The S9 switch can be in 3 different positions labeled numerically by an integer in the range 0 to 2. When the S9 switch is set to 0 (zero) there is no power on the LO1A LO signal (no synthesizer is selected) and the S9 switch is in a "undefined" state. When the S9 switch is set to 1 the LO signal is derived from the LO1A synthesizer. When the S9 switch is set to 2 the LO signal is derived from a second (user supplied?) synthesizer. Typically there will not be a synthesizer available other than the LO1A synthesizer.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s9 ≤ 2

Units: N/A

Example of Use: lo1.s9 = 1

lo1.s10

Description: The S10 switch determines which synthesizer is used to generate the LO1A LO signal. The S10 switch can be in 3 different positions labeled numerically by an integer in the range 0 to 2. When the S10 switch is set to 0 (zero) there is no power on the LO1A LO signal (no synthesizer is selected) and the S10 switch is in a "undefined" state. When the S10 switch is set to 1 the LO signal is derived from the LO1A synthesizer. When the S10 switch is set to 2 the LO signal is derived from a second (user supplied?) synthesizer. Typically there will not be a synthesizer available other than the LO1A synthesizer.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s10 ≤ 2

Units: N/A

Example of Use: lo1.s10 = 1

lo1.s11

Description: Either the output of the S3 switch (the "LO monitor signal") or the output of the Phase Cal generator can be used to provide a Test Tone for the receiver Test Tone output ports of the LO1 rack, via the S12, S13, S14 and S15 switches. The S11 switch determines whether the LO monitor signal or the Phase Cal signal is used as the Test Tone signal. When the S11 switch is in the "thru" state the LO monitor signal is terminated to ground and the Phase Cal signal is used for the Test Tone signal and is routed on to the S12 switch. When the S11 switch is in the "cross" state the Phase Cal signal is terminated to ground and the LO monitor signal is used for the Test Tone signal and is routed on to the S12 switch.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: "thru", "cross"

Units: N/A

Example of Use: lo1.s11 = "thru"

lo1.s12

Description: The S12 switch determines where the Test Tone signal is routed. The S12 switch can be in 5 different positions labeled numerically by an integer in the range 0 to 4. When the S12 switch is set to 0 (zero) there is no power on the Test Tone signal and the S12 switch is in a "undefined" state. When the S12 switch is set to 1 the Test Tone signal is terminated to ground. When the S12 switch is set to 2 the Test Tone signal is terminated to ground. When the S12 switch is set to 3 the Test Tone signal is routed to the Test Tone output ports 1-12 of the LO1 rack via the S14 and S15 switches. When the S12 switch is set to 4 the Test Tone signal is routed to the Test Tone output ports 13-18 of the LO1 rack via the S13 switch.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s12 ≤ 4

Units: N/A

Example of Use: lo1.s12 = 2

lo1.s13

Description: The S13 switch routes the Test Tone signal output from the S12 switch to one of the Test Tone output ports (13-18) of the LO1 rack. The S13 switch can be in 7 different positions labeled numerically by an integer in the range 0 to 6. When the S13 switch is set to 0 (zero) there is no power on the Test Tone signal and the S13 switch is in a "undefined" state. When the S13 switch is set to 1 the Test Tone signal is sent to the Test Tone output port 13 of the LO1 rack. When the S13 switch is set to 2 the Test Tone signal is sent to the Test Tone output port 14 of the LO1 rack. When the S13 switch is set to 3 the Test Tone signal is sent to the Test Tone output port 15 of the LO1 rack. When the S13 switch is set to 4 the Test Tone signal is sent to the Test Tone output port 16 of the LO1 rack. When the S13 switch is set to 5 the Test Tone signal is sent to the Test Tone output port 17 of the LO1 rack. When the S13 switch is set to 6 the Test Tone signal is sent to the Test Tone output port 18 of the LO1 rack.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s13 ≤ 6

Units: N/A

Example of Use: lo1.s13 = 2

lo1.s14

Description: The S14 switch routes the Test Tone signal output from the S12 switch to one of the Test Tone output ports (1-6) of the LO1 rack. The S14 switch can be in 7 different positions labeled numerically by an integer in the range 0 to 6. When the S14 switch is set to 0 (zero) there is no power on the Test Tone signal and the S14 switch is in a "undefined" state. When the S14 switch is set to 1 the Test Tone signal is sent to the Test Tone output port 1 of the LO1 rack. When the S14 switch is set to 2 the Test Tone signal is sent to the Test Tone output port 2 of the LO1 rack. When the S14 switch is set to 3 the Test Tone signal is sent to the Test Tone output port 3 of the LO1 rack. When the S14 switch is set to 4 the Test Tone signal is sent to the Test Tone output port 4 of the LO1 rack. When the S14 switch is set to 5 the Test Tone signal is sent to the Test Tone output port 5 of the LO1 rack. When the S14 switch is set to 6 the Test Tone signal is sent to the Test Tone output port 6 of the LO1 rack.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s14 ≤ 6

Units: N/A

Example of Use: lo1.s14 = 2

lo1.s15

Description: The S15 switch routes the Test Tone signal output from the S12 switch to one of the Test Tone output ports (7-12) of the LO1 rack. The S15 switch can be in 7 different positions labeled numerically by an integer in the range 0 to 6. When the S15 switch is set to 0 (zero) there is no power on the Test Tone signal and the S15 switch is in a "undefined" state. When the S15 switch is set to 1 the Test Tone signal is sent to the Test Tone output port 7 of the LO1 rack. When the S15 switch is set to 2 the Test Tone signal is sent to the Test Tone output port 8 of the LO1 rack. When the S15 switch is set to 3 the Test Tone signal is sent to the Test Tone output port 9 of the LO1 rack. When the S15 switch is set to 4 the Test Tone signal is sent to the Test Tone output port 10 of the LO1 rack. When the S15 switch is set to 5 the Test Tone signal is sent to the Test Tone output port 11 of the LO1 rack. When the S15 switch is set to 6 the Test Tone signal is sent to the Test Tone output port 12 of the LO1 rack.

One should consult the "cabling file" to determine how signals propagate between devices from a given output port to a given input port.

Values: integer

Allowed Range: $1 \leq$ lo1.s15 ≤ 6

Units: N/A

Example of Use: lo1.s15 = 2

4.4.2 Converter Rack

- conv.abselect
- conv.inputselect
- conv.outputselect
- conv.attenuator1
- conv.attenuator2

- conv.attenuator3
- conv.attenuator4
- conv.attenuator5
- conv.attenuator6
- conv.attenuator7
- conv.attenuator8
- conv.attenuator9
- conv.attenuator10
- conv.attenuator11
- conv.attenuator12
- conv.attenuator13
- conv.attenuator14
- conv.attenuator15
- conv.attenuator16
- conv.lofrequency
- conv.lolevel

conv.abselect

Description: The ConverterRack bank select switch (selects between Rack A and B's ConverterModules for the SpectralProcessor)

Values: A, B

Units: N/A

Example of Use: conv.abselect = A

conv.inputselect

Description: The Converter Input parameter selects between A and B inputs of each IF channel in the Converter Rack. These inputs are normally connected outputs of the optical fiber receiver modules.

Values: A, B

Max Size: 16

Units: N/A

Example of Use: conv.inputselect = A
conv.inputselect[1] = A
conv.inputselect[2] = B

conv.outputselect

Description: The Output Select parameter selects one of four output frequency ranges from each of the 16 IF channels in the Converter Rack. Selection 1 corresponds to the 150-2000 MHz output connector J3 (Sampler Filter); selection 2 to the 500-1000 MHz output connector J4 (VLBA DAR); selection 3 to four 150-550 MHz parallel outputs J5 (spectral processor), J6 (Converter Filter), J7 (spare), and J8 (spare); and selection 4 to the 150-2000 MHz output J9 (spare).

Values: 1, 2, 3, 4

Max Size: 16

Units: N/A

Example of Use: conv.outputselect = 1
conv.outputselect[1]=1
conv.outputselect[2]=3

conv.attenuator1

Description: The Attenuator1 parameter sets the attenuator values for the 1st IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator1 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator1 = 8.0

conv.attenuator2

Description: The Attenuator2 parameter sets the attenuator values for the 2nd IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator2 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator2 = 9.0

conv.attenuator3

Description: The Attenuator3 parameter sets the attenuator values for the 3rd IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator3 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator3 = 23.125

conv.attenuator4

Description: The Attenuator4 parameter sets the attenuator values for the 4th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator4 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator4 = 19.0

conv.attenuator5

Description: The Attenuator5 parameter sets the attenuator values for the 5th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator5 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator5 = 13.0

conv.attenuator6

Description: The Attenuator6 parameter sets the attenuator values for the 6th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator6 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator6 = 2.8

conv.attenuator7

Description: The Attenuator7 parameter sets the attenuator values for the 7th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator7 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator7 = 5.0

conv.attenuator8

Description: The Attenuator8 parameter sets the attenuator values for the 8th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator8 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator8 = 4.75

conv.attenuator9

Description: The Attenuator9 parameter sets the attenuator values for the 9th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator9 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator9 = 4.5

conv.attenuator10

Description: The Attenuator10 parameter sets the attenuator values for the 10th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator10 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator10 = 4.0

conv.attenuator11

Description: The Attenuator11 parameter sets the attenuator values for the 11th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator11 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator11 = 3.5

conv.attenuator12

Description: The Attenuator12 parameter sets the attenuator values for the 12th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator12 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator12 = 3.0

conv.attenuator13

Description: The Attenuator13 parameter sets the attenuator values for the 13th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator13 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator13 = 2.5

conv.attenuator14

Description: The Attenuator14 parameter sets the attenuator values for the 14th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: 0.0 $\leq$ conv.attenuator14 $\leq$ 38.875

Units: dB

Example of Use: conv.attenuator14 = 2.0

conv.attenuator15

Description: The Attenuator15 parameter sets the attenuator values for the 15th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: $0.0 \leq$ conv.attenuator15 ≤ 38.875

Units: dB

Example of Use: conv.attenuator15 = 1.5

conv.attenuator16

Description: The Attenuator16 parameter sets the attenuator values for the 16th IF channel in the Converter Rack. The attenuator value can be between 0 and 31.875 dB in 1/8th dB steps. Values are specified in floating point and the nearest available value is set. When the Converter Rack is used in connection with the autocorrelation GBT spectrometer these values may be set automatically by the sampler level balancing routine.

Values: float

Allowed Range: $0.0 \leq$ conv.attenuator16 ≤ 38.875

Units: dB

Example of Use: conv.attenuator16 = 1.0

conv.lofrequency

Description: The LO Frequency sets the frequency of each of 8 LO synthesizers in the Converter Rack. The output of each synthesiser is shared by two IF channels in this rack (often connected to two orthogonally polarized front-end channels to be observed at the same frequency). This is the second LO (LO2) in the GBT frequency conversion chain. Its range is 10500 to 18000 MHz, and it is used to conver frequencies in the 1 to 8 GHz passband of the optical fiber transceivers to a fixed second IF passband of 8500 to 10350 MHz.

Values: float

Max Size: 8

Units: MHz

Example of Use: conv.lofrequency = 13250.0
conv.lofrequency[1]=13260.0
conv.logfrequency[2]=13400.5

conv.lolevel

Description: The LO Level parameter sets the power level, in dBm, for each of the 8 frequency synthesizers (LO2) in the Converter Rack.

Values: float

Max Size: 8

Units: dBm

Example of Use: conv.lolevel = 10.0
conv.lolevel[1]=10.0

4.4.3 IF Rack

- ifrack.laserpower
- ifrack.attenuator
- ifrack.analogpowerlevel
- ifrack.balance
- ifrack.balanceselect
- ifrack.noisebandwidth
- ifrack.s1
- ifrack.s2
- ifrack.s3
- ifrack.s4
- ifrack.s5
- ifrack.s6
- ifrack.s7
- ifrack.s8
- ifrack.s9
- ifrack.s10
- ifrack.s11
- ifrack.s12
- ifrack.filterselect
- ifrack.laserautolevelcontrol

ifrack.laserpower

Description: Laser power switches for laser transmitters.

Values: swOff, swOn

Max Size: 8

Units: N/A

Example of Use: ifrack.laserpower = swOn

ifrack.attenuator

Description: 0dB - 31dB attenuators for laser transmitters.

Values: float

Max Size: 8

Units: dB

Example of Use: ifrack.attenuator[1] = 3.0
ifrack.attenuator[5] = 5.3

ifrack.analogpowerlevel

Description: Sets the target levels in volts for the laser transmitters.

Values: float

Max Size: 8

Units: volts

Example of Use: ifrack.analogpowerlevel = 1.0

ifrack.balance

Description: Initiates balancing of the power levels for selected laser transmitters when changed from a zero to non-zero integer.

Values: YES, NO

Units: N/A

Example of Use: ifrack.balance = YES

ifrack.balanceselect

Description: Selects laser transmitters for balancing.

Values: On , Off

Max Size: 8

Units: N/A

Example of Use: ifrack.balanceselect[1] = On
ifrack.balanceselect[2] = Off

ifrack.noisebandwidth

Description: Noise source bandwidth control.

Values: broadband, narrowband

Units: N/A

Example of Use: ifrack.noisebandwidth = narrowband

ifrack.s1

Description: Selects one of IF-1 thru IF-8.

Values: integer

Allowed Range: $1 \leq$ ifrack.s1 ≤ 8

Units: N/A

Example of Use: ifrack.s1 = 1

ifrack.s2

Description: Selects one of IF-9 thru IF-16.

Values: integer

Allowed Range: $1 \leq$ ifrack.s2 ≤ 8

Units: N/A

Example of Use: ifrack.s2 = 2

ifrack.s3

Description: Selects one of IF-17 thru IF-24.

Values: integer

Allowed Range: $1 \leq$ ifrack.s3 ≤ 8

Units: N/A

Example of Use: ifrack.s3 = 3

ifrack.s4

Description: Selects one of IF-25 thru IF-32.

Values: integer

Allowed Range: $1 \leq$ ifrack.s4 ≤ 8

Units: N/A

Example of Use: ifrack.s4 = 4

ifrack.s5

Description: Selects one of IF-33 thru IF-40.

Values: integer

Allowed Range: $1 \leq$ ifrack.s5 ≤ 8

Units: N/A

Example of Use: ifrack.s5 = 5

ifrack.s6

Description: Selects one of IF-41 thru IF-48.

Values: integer

Allowed Range: $1 \leq$ ifrack.s6 ≤ 8

Units: N/A

Example of Use: ifrack.s6 = 6

ifrack.s7

Description: Selects one of IF-49 thru IF-56.

Values: integer

Allowed Range: $1 \leq$ ifrack.s7 ≤ 8

Units: N/A

Example of Use: ifrack.s7 = 7

ifrack.s8

Description: Selects one of IF-57 thru IF-64.

Values: integer

Allowed Range: $1 \leq$ ifrack.s8 ≤ 8

Units: N/A

Example of Use: ifrack.s8 = 8

ifrack.s9

Description: Crosses or passes outputs of S1 and S2.

Values: thru, cross

Units: N/A

Example of Use: ifrack.s9 = thru

ifrack.s10

Description: Crosses or passes outputs of S3 and S4.

Values: thru, cross

Units: N/A

Example of Use: ifrack.s10 = cross

ifrack.s11

Description: Crosses or passes outputs of S5 and S6.

Values: thru, cross

Units: N/A

Example of Use: ifrack.s11 = thru

ifrack.s12

Description: Crosses or passes outputs of S7 and S8.

Values: thru, cross

Units: N/A

Example of Use: ifrack.s12 = cross

ifrack.filterselect

Description: Selects among 8 bandpass filter options.

Values: pass_all, pass_2990_3010, pass_2960_3040, pass_2840_3160, pass_2360_3640, pass_5960_6040, pass_5840_6160, pass_5360_6640

Max Size: 8

Units: N/A

Example of Use: ifrack.filterselect[1] = pass_2960_3040
ifrack.filterselect[2] = pass_2840_3160

ifrack.laserautolevelcontrol

Description: Laser power switches for laser automatic level control.

Values: swOn, swOff

Max Size: 8

Units: N/A

Example of Use: ifrack.laserautolevelcontrol[1] = swOn
ifrack.laserautolevelcontrol[2] = swOff

4.4.4 Analog Filter Rack

- algfilt.sginput
- algfilt.sgfilter

- algfilt.cffilter
- algfilt.oneppsenable

algfilt.sginput

Description: Selects input from 1-8 GHz Converter Modules
Values: 1, 2, 3, 4
Max Size: 8
Units: N/A
Example of Use: algfilt.sginput = 1

algfilt.sgfilter

Description: Selects output filter.
Values: wide, narrow, spare, external
Max Size: 8
Units: N/A
Example of Use: algfilt.sgfilter = wide

algfilt.cffilter

Description: Selects filter.
Values: wide, narrow, spare, external
Max Size: 16
Units: N/A
Example of Use: algfilt.cffilter = narrow

algfilt.oneppsenable

Description: Controls 1PPS synchronization.
Values: On , Off
Units: N/A
Example of Use: algfilt.oneppsenable = On

4.5 Front Ends

4.5.1 rx Keyword Prefix

Not yet implemented in GO.

4.5.2 Prime Focus 1 Receiver

Not yet implemented in GO.

4.5.3 Prime Focus 2 Receiver

Not yet implemented in GO.

4.5.4 1 To 2 GHz (21 cm) Receiver

- rx1to2.yrcpunoiseswctrl
- rx1to2.xlcpunoiseswctrl
- rx1to2.cpulocalpwrsw
- rx1to2.loorhicalsel
- rx1to2.xlexttomcbctrlsel
- rx1to2.yrexttomcbctrlsel
- rx1to2.rightiffilterswitch
- rx1to2.leftiffilterswitch
- rx1to2.lincircphaseshift
- rx1to2.polarizationselect
- rx1to2.xferswitch
- rx1to2.xferswctlmode

rx1to2.yrcpunoiseswctrl

Description: Controls the state of the cal switch for the YR polarization when using MCB control of the cal signal.

Values: "swOn", "swOff"

Units: N/A

Example of Use: rx1to2.yrcpunoiseswctrl=swOn

rx1to2.xlcpunoiseswctrl

Description: Controls the state of the cal switch for the XL polarization when using MCB control of the cal signal.

Values: rx1to2.xlcpunoiseswctrl=swOff

Units: N/A

Example of Use: "swOn", "swOff"

rx1to2.cpulocalpwrsw

Description: Controls the state of the power supply which powers the Low Cal noise diode.

Values: rx1to2.cpulocalpwrsw=swOn

Units: N/A

Example of Use: "swOn", "swOff"

rx1to2.loorhicalsel

Description: Controls the selection of either the Low or High Cal noise diode.

Values: rx1to2.loorhicalsel = lowCal

Units: N/A

Example of Use: "lowCal", "highCal"

rx1to2.xlexttomcbctrlsel

Description: Sets the XL polariztion cal control mode to external (manual) or mcb control.

Values: rx1to2.xlexttomcbctrlsel = ctlMcb

Units: N/A

Example of Use: "ctlExt", "ctlMcb"

rx1to2.yrexttomcbctrlsel

Description: Sets the YR polarization cal control mode to external (manual) or mcb control.

Values: rx1to2.yrexttomcbctrlsel = ctlExt

Units: N/A

Example of Use: "ctlExt", "ctlMcb"

rx1to2.rightiffilterswitch

Description: This parameter selects the YR polarization bandpass filter. The following integer values are acceptable:

1—-> Filter Bandwidth = 1.1 - 1.8 GHz
2—-> Filter Bandwidth = 1.6 - 1.75 GHz
3—-> Filter Bandwidth = 1.3 - 1.45 GHz
4—-> Filter Bandwidth = 1.1 - 1.45 GHz
5—-> Spare

Values: integer

Allowed Range: $0 \leq$ rx1to2.rightiffilterswitch ≤ 6

Units: N/A

Example of Use: rx1to2.rightiffilterswitch = 2

rx1to2.leftiffilterswitch

Description: This parameter selects the XL polarization bandpass filter. The following integer values are acceptable:

1—-> Filter Bandwidth = 1.1 - 1.8 GHz
2—-> Filter Bandwidth = 1.6 - 1.75 GHz
3—-> Filter Bandwidth = 1.3 - 1.45 GHz
4—-> Filter Bandwidth = 1.1 - 1.45 GHz
5—-> Spare

Values: integer

Allowed Range: $0 \leq$ rx1to2.leftiffilterswitch ≤ 6

Units: N/A

Example of Use: rx1to2.leftiffilterswitch = 3

rx1to2.lincircphaseshift

Description: This parameter sets the polarization hybrid phase shifter phase value. The range of possible values are 0.7 ¡ phase ¡ 89.3 degrees.

Values: float

Allowed Range: $0.7 \leq$ rx1to2.lincircphaseshift ≤ 89.3

Units: degrees

Example of Use: rx1to2.lincircphaseshift = 40.5

rx1to2.polarizationselect

Description: Controls the selection of linear or circular polarization.

Values: rx1to2.polarizationselect = Circular

Units: N/A

Example of Use: “Linear”, “Circular”

rx1to2.xferswitch

Description: Sets the polarization transfer switch to thru or crossed.

Values: rx1to2.xferswitch = tsCrossed

Units: N/A

Example of Use: “tsThru”, “tsCrossed”

rx1to2.xferswctlmode

Description: Sets the control mode of the transfer switch to external (manual) or mcb control.

Values: rx1to2.xferswctlmode = ctlExt

Units: N/A

Example of Use: “ctlExt”, “ctlMcb”

4.5.5 2 To 3 GHz (11 cm) Receiver

Not yet implemented in GO.

4.5.6 4 To 6 GHz (6 cm) Receiver

Not yet implemented in GO.

4.5.7 8 To 10 GHz (3 cm) Receiver

Not yet implemented in GO.

4.5.8 12 To 18 GHz (2 cm) Receiver

- rx12to18.callcp
- rx12to18.calrcppower
- rx12to18.callcppower
- rx12to18.calctrl
- rx12to18.rcp12
- rx12to18.lcp12
- rx12to18.rcp1iffilterswitch
- rx12to18.lcp1iffilterswitch
- rx12to18.beamctrl

rx12to18.callcp

Description: Controls the state of the CAL LCP switch. The switchcan be either ”on” or ”off”. Note that this keyword is only used if cal_ctrl is set to ”manual”.

Values: “on”, “off”

Units: N/A

Example of Use: rx12to18.cal_lcp = ”off”

rx12to18.calrcppower

Description: Power control for the RCP CAL for the 1.5 cm (12-18 GHz) receiver.The power supply can be either ”on” or ”off”.

Values: “on”, “off”

Units: N/A

Example of Use: rx12to18.cal_rcp_power = ”off”

rx12to18.callcppower

Description: Power control for the LCP CAL for the 1.5 cm (12-18 GHz) receiver. The power supply can be either ”on” or ”off”.

Values: “on”, “off”

Units: N/A

Example of Use: rx12to18.cal_lcp_power = ”off”

rx12to18.calctrl

Description: Controls the operation of CAL switches for both LCP and RCP forthe 1.5 cm (12-18 GHz) receiver. CAL is short for calibrationand corresponds to noise injected into the system for calibrationpurposes. These switches can be set either manually or be underexternal control as defined by cal_ctrl.

Values: “external”, “manual”

Units: N/A

Example of Use: rx12to18.cal_ctrl = ”external”

rx12to18.rcp12

Description: Controls the state of the RCP switch for the 18-22 GHz sectionof the 1.5 cm receiver between feeds 1 and 2. The switch caneither be through or crossed. Note that this keyword is onlyused if beam_ctrl is set to ”manual”.

Values: “thru”, “crossed”

Units: N/A

Example of Use: rx12to18.rcp12 = ”crossed”

rx12to18.lcp12

Description: Controls the state of the LCP switch for the 18-22 GHz sectionof the 1.5 cm receiver between feeds 1 and 2. The switch caneither be through or crossed. Note that this keyword is onlyused if beam_ctrl is set to ”manual”.

Values: “thru”, “crossed”

Units: N/A

Example of Use: rx12to18.lcp12 = ”crossed”

rx12to18.rcp1iffilterswitch

Description: Controls the state of the RCP switch for the 18-22 GHz sectionof the 1.5 cm receiver between feeds 1 and 2. The switch caneither be through or crossed. Note that this keyword is onlyused if beam_ctrl is set to ”manual”.

Values: “3000/3500”, “3000/500”

Units: N/A

Example of Use: rx12to18.rcp1iffilterswitch = ”crossed”

rx12to18.lcp1iffilterswitch

Description: Controls the state of the LCP switch for the 18-22 GHz sectionof the 1.5 cm receiver between feeds 1 and 2. The switch caneither be through or crossed. Note that this keyword is onlyused if beam_ctrl is set to ”manual”.

Values: “3000/3500”, “3000/500”

Units: N/A

Example of Use: rx12to18.lcp1iffilterswitch = ”crossed”

rx12to18.beamctrl

Description: Controls the operation of all four beam switches for the 1.5 cm(18-26 GHz) receiver. These switches can either be set manuallyor be under external control as defined by beam_ctrl. Thisreceiver consists of two separate dual-polarization feeds. There are a total of two beam switchscorresponding to switching between the LCP and RCP signals of thetwo 12-18 GHz feeds. The switches are called rcp12 and lcp12.beam_ctrl, which simultaneously controls bothswitches,can be in either manual or external control.

Values: “external”, “manual”

Units: N/A

Example of Use: rx12to18.beam_ctrl = ”external”

4.5.9 18 To 26 GHz (1 cm) Receiver

- rx18to26.cal_rcp
- rx18to26.cal_lcp
- rx18to26.cal_rcp_power
- rx18to26.cal_lcp_power
- rx18to26.cal_ctrl
- rx18to26.rcp12
- rx18to26.lcp12
- rx18to26.rcp34
- rx18to26.lpc34
- rx18to26.beam_ctrl

rx18to26.cal_rcp

Description: Controls the state of the CAL RCP switch. The switch can be either “on” or “off”. Note that this keyword is only used if cal_ctrl is set to “manual”.

Values: on, off

Units: N/A

Example of Use: rx18to26.cal_rcp = ”off”

rx18to26.cal_lcp

Description: Controls the state of the CAL LCP switch. The switch can be either "on" or "off". Note that this keyword is only used if cal_ctrl is set to "manual"

Values: on, off

Units: N/A

Example of Use: rx18to26.cal_lcp = "off"

rx18to26.cal_rcp_power

Description: Power control for the RCP CAL for the 1 cm (18-26 GHz) receiver. The power supply can be either "on" or "off".

Values: on, off

Units: N/A

Example of Use: rx18to26.cal_rcp_power = "off"

rx18to26.cal_lcp_power

Description: Power control for the LCP CAL for the 1 cm (18-26 GHz) receiver. The power supply can be either "on" or "off".

Values: on, off

Units: N/A

Example of Use: rx18to26.cal_lcp_power = "off"

rx18to26.cal_ctrl

Description: Controls the operation of CAL switches for both LCP and RCP for the 1 cm (18-26 GHz) receiver. CAL is short for calibration and corresponds to noise injected into the system for calibration purposes. These switches can be set either manually or be under external control as defined by cal_ctrl.

Values: manual, external

Units: N/A

Example of Use: rx18to26.cal_ctrl = "external"

rx18to26.rcp12

Description: Controls the state of the RCP switch for the 18-22 GHz section of the 1 cm receiver between feeds 1 and 2. The switch can either be through or crossed. Note that this keyword is only used if beam_ctrl is set to "manual".

Values: thru, cross

Units: N/A

Example of Use: rx18to26.rcp12 = "thru"

rx18to26.lcp12

Description: Controls the state of the LCP switch for the 18-22 GHz section of the 1 cm receiver between feeds 1 and 2. The switch can either be through or crossed. Note that this keyword is only used if beam_ctrl is set to "manual".

Values: thru, cross

Units: N/A

Example of Use: rx18to26.lcp12 = "thru"

rx18to26.rcp34

Description: Controls the state of the RCP switch for the 22-26 GHz section of the 1 cm receiver between feeds 3 and 4. The switch can either be through or crossed. Note that this keyword is only used if beam_ctrl is set to "manual".

Values: thru, cross

Units: N/A

Example of Use: rx18to26.rcp34 = "thru"

rx18to26.lpc34

Description: Controls the state of the LCP switch for the 22-26 GHz section of the 1 cm receiver between feeds 3 and 4. The switch can either be through or crossed. Note that this keyword is only used if Kbeam_ctrl is set to "manual".

Values: thru, cross

Units: N/A

Example of Use: rx18to26.lpc34 = "thru"

rx18to26.beam_ctrl

Description: Controls the operation of all four beam switches for the 1 cm (18-26 GHz) receiver. These switches can either be set manually or be under external control as defined by beam_ctrl. This receiver consists of four separate dual-polarization feeds. Feeds 1 and 2 can be tuned between 18-22 GHz while feeds 3 and 4 can be tuned between 22-26 GHz. There are a total of four beam switches corresponding to switching between the LCP and RCP signals of the two 22-18 GHz feeds and between the LCP abd RCP signals of the two 22-26 GHz feeds. The switches are called rcp12, lcp12, rcp34, and lcp34. beam_ctrl, which simultaneously controls all four switches, can be in either manual or external control.

Values: manual, external

Units: N/A

Example of Use: rx18to26.beam_ctrl = "external"

4.5.10 40 To 52 GHz (6 mm) Receiver

Not yet implemented in GO.

4.6 Backends

4.6.1 Digital Continuum Receiver

- dcr.bank
- dcr.switchperiod
- dcr.integrationtime
- dcr.channel

dcr.bank

Description: The DCR has two banks of inputs identified as Bank A and Bank B. Each bank has 16 input channels each. The Bank is normally defaulted to the correct one for the front-end in use, but check with the telescope operator or receiver engineer, if you are uncertain.

Values: A, B

Units: N/A

Example of Use: dcr.bank = A

dcr.switchperiod

Description: The Switch Period specifies the time in seconds of a full switch cycle. The Integration Time must be an integer number of Switch Periods and will be changed to the nearest value when a new Switch Period is entered.

Values: float

Units: seconds

Example of Use: dcr.switchperiod = 0.2

dcr.integrationtime

Description: Integration Time is the time of one recorded data sample. It is automatically adjusted to be an integer number of Switch Periods. One data sample will contain a separate data integration for each phase in the switching cycle.

Values: float

Units: seconds

Example of Use: dcr.integrationtime = 0.2

dcr.channel

Description: Any of 16 input channels may be activated by toggling its button to”on” as indicated by its square dot being dark.

Values: On, Off

Max Size: 16

Units: N/A

Example of Use: dcr.channel[1] = On
dcr.channel[3]=Off

4.6.2 Spectral Processor

- sp.configuration
- sp.bandwidth
- sp.integrationtime
- sp.atodlevelmode
- sp.balance
- sp.multipliermode
- sp.taper
- sp.caldutycycle
- sp.calphase
- sp.dispersionmeasure
- sp.polycodatfile
- sp.pulseperiod
- sp.pulseoffset
- sp.sampletime
- sp.calpulsarstate
- sp.switchperiod
- sp.addpolarizations
- sp.datashift
- sp.deadfftstagea
- sp.deadfftstageb
- sp.fastformat
- sp.randomoffset
- sp.rawdata
- sp.rackstatus
- sp.taperoffset
- sp.atodinputlevel
- sp.caltemperature
- sp.iffrequency
- sp.ifsideband
- sp.rfsideband
- sp.skyfrequency
- sp.cliplevel

- sp.excise
- sp.fasttimeconst
- sp.slowtimeconst
- sp.threshold
- sp.iflosource
- sp_stor.tapeid
- sp_stor.tapedirect
- sp_stor.eoftrigger
- sp_stor.eofdelay
- sp_stor.eofcontrol
- sp_stor.writefile
- sp_stor.filecontrol
- sp_stor.sampleinterval
- sp_stor.filename
- sp_stor.collate
- sp_stor.log
- sp_stor.flushbuffer

sp.configuration

Description: The spectral processor Configuration selects from the four available combinations of number of IF's and number of spectral channels per IF

Values: "2x1024 IFxCh", "4x512 IFxCh", "4x256 IFxCh", "8x256 IFxCh"

Units: N/A

Example of Use: sp.configuration = "2x1024 IFxCh"

sp.bandwidth

Description: The Bandwidth parameter selects from ten available bandwidths per IF channel from 40 MHz down to 78 kHz. The bandwidth upper limit is 20 MHz for Configurations of 4 IF's and 10 MHz for the 8-IF Configuration

Values: " 40 MHz ", " 20 MHz ", " 10 MHz ", " 5 MHz ", "2.5 MHz ", "1.25 MHz ", "0.625 MHz", "0.312 MHz", "0.156 MHz", "0.078 MHz"

Units: N/A

Example of Use: sp.bandwidth = "1.25 MHz "

sp.integrationtime

Description: This parameter is the Integration Time for one data record written to disk. Partial integrations at the end of a scan are normally discarded so you will want the Scan Length to be an integer number of Integration Times plus a second or two to be sure that the last integration is completed. Long integrations risk greater loss of data from a corrupted integration while shorter integrations fill up the disk faster.

Values: float

Units: seconds

Example of Use: sp.integrationtime = 10

sp.atodlevelmode

Description: The A/D Leveling Mode determines when the attenuator settings are changed. The ScanStart selection causes the attenuators to be set just before each scan starts, if the Balance selection is 'yes'

Values: Immediate , "Scan Start"

Units: N/A

Example of Use: sp.atodlevelmode = "Scan Start"

sp.balance

Description: Balance specifies whether or not the input levels to the A/D convertors are set at the beginning of a scan (or immediately if selected by the A/D Level Mode).

Values: YES, NO

Units: N/A

Example of Use: sp.balance = NO

sp.multipliermode

Description: Multiplier Mode sets the configuration of the multiplier following the output of each complex FFT and real correction. The three possibilities are to square the output spectral values, cross multiply the outputs from racks A and B for polarization or other cross-correlation work, or do both if the bandwidth permits the multiplier to run twice as fast as the FFT engine.

Values: Square , Cross , SqrCross

Units: N/A

Example of Use: sp.multipliermode = Square

sp.taper

Description: The Taper specifies the weighting function to be applied to the input A/D amplitude-vs-time series before it is transformed. In spectral processor hardware memos the taper function is called a window. The Box taper weights all input samples equally, Cosine uses a half-cosine cycle as a weighting function, and Halfbox uses unity weight in the center half of the samples and zero for the first and last quarter (mainly for test purposes). The taper affects the spectrometer resolution, spectral channel 'sidelobes', and, to some extent, the spectrometer sensitivity.

Values: Box , Cosine , Halfbox

Units: N/A

Example of Use: sp.taper = Box

sp.caldutycycle

Description: Cal Duty Cycle specifies the duration of the pulsed calibration signal. In the pulsar timing mode this is the fraction of the pulsar period. In dedispersion mode the cal fires at a one-second interval, and the Cal Duty Cycle is the fraction of this interval. The value must be between 0.0 and 1.0. This parameter is ignored in other modes. It is used in conjuction with Cal Phase

Values: float

Allowed Range: $0.0 \leq$ sp.caldutycycle ≤ 1.0

Units: fractional phase of pulse period

Example of Use: sp.caldutycycle = 0.1

sp.calphase

Description: Cal Phase specifies the start time of the calibration pulse. In the pulsar timing mode, Cal Phase specifies the start phase with respect to the beginning the pulse window in fraction of a pulse period. In dedispersion mode the cal fires at a one-second interval, and the Cal Phase is the fraction of this interval with respect to the start of the scan. The value must be between 0.0 and 1.0. This parameter is ignored in other modes. It is used in conjuction with Cal Duty Cycle.

Values: float

Allowed Range: $0.0 \leq$ sp.calphase ≤ 1.0

Units: fractional range of pulse period

Example of Use: sp.calphase = 0.9

sp.dispersionmeasure

Description: Dispersion Measure is used for computing start time offsets in pulsar-synchronous spectral processor modes so that the pulse is placed in the data location specified by Pulse Offset. This parameter is also used in pulsar dedispersion modes to configure the accumulator map. The unit is parsecs/cm**3. When a standard format Pulsar Name in entered the dispersion measure value will be picked up from the polyco.dat file.

Values: float

Allowed Range: $0.0 \leq$ sp.dispersionmeasure $\leq \infty$

Units: parsecs/cm$^{-}3$

Example of Use: sp.dispersionmeasure = 23.6

sp.polycodatfile

Description: The Pulse Coefficients File is the name of the file that contains pulse frequency (1/period) coefficients for the pulsars to be observed. This file must be prepared before an observing run using the 'tempo' program. The output of 'tempo' is an ascii file, called polyco.dat, which may be used directly or converted to a binary file with the 'polybinary' program for faster execution at the beginning of each scan. File names with the .bin suffix are assumed to be binary. Files with all other suffixes are assumed to be ascii.

Values: filename

Units: N/A

Example of Use: sp.polycodatfile = "/users/joesmoe/polyco.dat"

sp.pulseperiod

Description: The Pulse Period is the period of the observed pulsar used to synchronously average the data over many pulse periods. This value is usually determined from the pulse frequency coefficients in the polyco file, as specified by the Pulsar Name, plus a doppler correction. The pulse period may be specified directly, but it will not be updated for changing doppler shift during a scan as it is when taken from the coefficients file.

Values: float

Units: seconds

Example of Use: sp.pulseperiod = 0.734

sp.pulseoffset

Description: Pulse Offset specifies the offset of a pulse from the center of the pulse period window in a pulsar-synchronous accumulation mode. The offset is in fraction of a pulse period between -1.0 and 1.0. To the accuracy of the predicted pulse arrival time from the polyco file, the scan start time is adjusted to place the pulse in the selected accumulation window phase. In many cases the pulse phase has drifted since the last update of the data used in the polyco calculation so an initial observation may be required to get the current pulse offset.

Values: float

Allowed Range: $-1.0 \leq$ sp.pulseoffset ≤ 1.0

Units: fractional phase of pulse period

Example of Use: sp.pulseoffset = 0.0

sp.sampletime

Description: Sample Time is the accumulation time, in seconds, for each time bin in the continuous dedispersion mode of the spectral processor. This time will be adjusted to an integer number of FFT cycle times

Values: float

Units: seconds

Example of Use: sp.sampletime = 0.002

sp.calpulsarstate

Description: In pulsar timing and dedispersion modes, this button simply specifies whether the calibration signal should be pulsed with the Cal Duty Cycle and Cal Phase specified.

Values: On , Off

Units: N/A

Example of Use: sp.calpulsarstate = Off

sp.switchperiod

Description: Switch Period is the period, in seconds, of the cal and sig/ref switching cycle. Since this period must be an integer number of FFT cycle times, the actual switch period will be adjusted to the nearest possible value.

Values: float

Units: seconds

Example of Use: sp.switchperiod=1.0

sp.addpolarizations

Description: The Add Polarization flag turns on a requested software patch in the pulsar timing mode of the spectral processor to add spectra from two polarizations together in the hardware accumulator. This reduces the data rate into observer-supplied data acquisition equipment. It assumes that one polarization is hooked to IF A1 (B1) and the other to A2 (B2).

Values: YES, NO

Units: N/A

Example of Use: sp.addpolarizations = NO

sp.datashift

Description: Data Shift determines the bit range of the conversion of 32-bit data from the FFT engine to 16-bit data into the square/cross multiplier stage. In special cases where FFT output noise quantization might be a problem, such as low IF input levels, the data word shift may be increased. The default value is 2 bits, which means that the highest 2 bits (sign-extended) are dropped from the FFT output word. Since the FFT output is in the voltage domain, 2 bits is equivalent to the highest 4 bits in the power product. Available values for data shift are 2, 3, 4, and 5. The higher values produce larger power spectrum output values for a given FFT input level.

Values: integer

Units: bits

Example of Use: sp.datashift = 2

sp.deadfftstagea

Description: Dead FFT Stage is a list of statuses of 11 hardware FFT stages. If a stage is dead, it may be flagged here so that it is switched out of the system. There is one completely spare stage. The numbers of stages required in the various configurations are:
2 IF x 1024 ch: 10 (1 spare)
4 IF x 512 ch: 9 (2 spares)
4 IF x 256 ch: 8 (3 spares)
8 IF x 256 ch: 8 (3 spares)

Values: integer

Units: N/A

Example of Use: sp.deadfftstagea = 5

sp.deadfftstageb

Description: Dead FFT Stage is a list of statuses of 11 hardware FFT stages. If a stage is dead, it may be flagged here so that it is switched out of the system. There is one completely spare stage. The numbers of stages required in the various configurations are
2 IF x 1024 ch: 10 (1 spare)
4 IF x 512 ch: 9 (2 spares)
4 IF x 256 ch: 8 (3 spares)
8 IF x 256 ch: 8 (3 spares)

Values: integer

Units: N/A

Example of Use: sp.deadfftstageb = 3

sp.fastformat

Description: The Fast Format flag controls whether 32-bit or 16-bit words are transferred out of the accumulator. The 16-bit words are the least significant accumulator bytes so this word size is useful only for short integrations where the word does not overflow. This option has been used mainly with high speed dedispersed data streams, particularly in connection with the Raw Data parameter, but it works in other modes.

Values: YES, NO

Units: N/A

Example of Use: sp.fastformat = YES

sp.randomoffset

Description: In pulsar spectral processor modes the start time is advanced by a randomly generated time equal to a fraction of one time bin to reduce any timing bias due to time bin quantization. This offset may be turned off with the Random Offset flag.

Values: YES, NO

Units: N/A

Example of Use: sp.randomoffset = YES

sp.rawdata

Description: The Raw Data selection is used to bypass data formatting so that accumulator data is transferred directly to disk with a minimum of processing. This option was added mainly for high speed sampling in the pulsar dedispersion mode, but it works in other modes. In the dedispersion mode the value of RD may be set to select any 8, 16, or 32-bit portion of the samples from the accumulator. The choices are
Off Raw data turned off (default)
B1 Save only the least significant byte.
B2 Save only the second least significant byte.
B3 Save only the third least significant byte.
B4 Save only the most significant byte.
W1 Save the least significant 16-bit word, bytes 1&2
W2 Save the middle 16-bit word, bytes 2&3
W3 Save the most significant 16-bit word, bytes 3&4
Full Save the full accumulator 16 or 32 bit word as set by the Fast Format parameter.Values B3, B4, W2, and W3 work only when the full 32-bit word is transferred from the accumulator (Fast Format off). Only Raw Data selections 'Off' and 'Full' work in the non-dedispersion modes.

Values: Off , B1 , B2 , B3 , B4 , W1 , W2 , W3 , Full

Units: N/A

Example of Use: sp.rawdata = Full

sp.rackstatus

Description: The spectral processor is actually two independent FFT engines which, in principle, may be run independently. Since this independence has never been used for observing, it is not provided for in this observer interface, except for Taper Offset. If glish commands are used to decouple the two FFT engines (racks), this status will be shown in the Rack Status field. The possibilities are both racks off, rack A only, rack B only, and both racks on (may or may not be being used independently).

Values: Off , "Rack A", "Rack B", Both

Units: N/A

Example of Use: sp.rackstatus = Both

sp.taperoffset

Description: The spectral processor is actually two separate FFT engines which, in principle, may be run independently. The Taper Offset flag selects whether the data sampling for the time series to be transformed in racks A and B are started together or are offset by half of a series length. If both racks look at the same IF signal, the offset provides inhanced sensitivity when spectra from the same IF are added together. The penalty is that there are half as many spectral channels to spread around the frequency dimensions. This option cannot be used in the 'Cross' or 'Square/Cross' Multiplier Mode since the inputs to racks A and B must be sampled synchronously.

Values: YES, NO

Units: N/A

Example of Use: sp.taperoffset = NO

sp.atodinputlevel

Description: A/D Input Level sets the noise level at the input to each A/D convertor. The unit is the number of quantization intervals per noise rms level, normally between about 1.0 and 4.0. Lower numbers give more large signal handling room in the A/D but compromize sensitivity and baseline stability because of quantization effects. The IF attenuators will be set to produce an input level within about 0.7 dB of the one specified

Values: float

Max Size: 8

Units: quantization intervals per noise rms level

Example of Use: sp.atodinputlevel[1] = 2.5
sp.atodinputlevel[5]=2.5

sp.caltemperature

Description: Cal Temperature is the calibration source intensity in Kelvins. This parameter has no direct effect on the operation of the spectral processor. Its value is passed on to the data header for later data reduction.

Values: float

Max Size: 8

Units: Kelvins

Example of Use: sp.caltemperature[1] = 1.62
sp.caltemperature[5] = 1.67

sp.iffrequency

Description: The I.F. Frequencies are the center frequencies of IF passbands. After taking into account the baseband offset and other conversions in the IF drawer, these parameters set the frequencies of the synthesizers in the IF drawers or the frequency of the high resolution synthesizer if an external IF LO is selected. The actual passband center frequency may be slighly different from the values specified because the resolution of the internal synthesizers is 10 kHz. The actual IF center frequencies are recorded with the data. The center IF frequency corresponds to the center of channel N/2+1, where N is the number of channels in the spectrum, and the first channel number is 1.

Values: float

Max Size: 8

Units: MHz

Example of Use: sp.iffrequency[1] = 250.0
sp.iffrequency[5]=250.0

sp.ifsideband

Description: IF Sideband selects the active single-sideband convertor sideband. With upper sideband, increasing frequency at the IF corresponds to increasing frequency at baseband. Lower sideband, produces decreasing frequency at baseband with increasing frequency in the IF passband. The RF Sideband parameter, as well as this parameter, affects the frequency transformation from sky to baseband frequency. In the final spectrum increasing channel number corresponds to increasing baseband frequency.

Values: Upper, Lower

Max Size: 8

Units: N/A

Example of Use: sp.ifsideband[1] = Upper
sp.ifsideband[5] = Lower

sp.rfsideband

Description: The RF Sideband specifies the sky-to-IF-passband frequency inversion for each IF channel. This is required in the hardware dedispersion modes, along with the IF Sideband, to determine the direction of dispersion. In other modes, these parameters are used to interpret the frequency direction of the output spectra.

Values: Upper, Lower

Max Size: 8

Units: N/A

Example of Use: sp.rfsideband[1] = Upper
sp.rfsideband[5] = Lower

sp.skyfrequency

Description: The Sky Frequency specifies the center of the passband being observed. This parameter is required in the hardware dedispersion modes to convert dispersion measure into a time delay. In other modes, it is used to interpret the frequency scale of the output spectra.

Values: float

Max Size: 8

Units: MHz

Example of Use: sp.skyfrequency[1] = 1420.4058
sp.skyfrequency[5] = 1720.09

sp.cliplevel

Description: Clipping Level sets the maximum level that can be instantaneously applied to the input of the slow baseline integrator in the RFI detector. This clipping prevents severe over-charging and, hence, slow recovery of this integrator. This parameter is specified in volts applied to the clipper diode. The maximum values is 10 volts, which is the default.

Values: float

Allowed Range: $0.0 \leq$ sp.cliplevel ≤ 10.0

Units: volts

Example of Use: sp.cliplevel = 10.0

sp.excise

Description: Excise turns real-time RFI excision on or off. RFI excision is based on total power threshold detection in each IF channel with the detector parameters being Threshold, Clip Level, and Fast and Slow Time Constants.

Values: YES, NO

Units: N/A

Example of Use: sp.excise = NO

sp.fasttimeconst

Description: Fast Time Constant specifies the response time of the total power RFI threshold detector. The optimum value depends on the nature of the RFI but is usually roughly equal to the characteristic pulse length of the interference.

Values: 1us , 3us , 10us , 30us , 100us, 300us, 1ms , 3ms , 10ms

Units: N/A

Example of Use: sp.fasttimeconst = ”3us”

sp.slowtimeconst

Description: Slow Time Constant sets the response time of the baseline comparison level for the total power RFI threshold detector. The optimum value depends on the nature of the RFI and is usually more than 10 times the characteristic time scale of the transient interference. RFI detection occurs when the fast-time-constant detector output exceeds the slow-time-constant output by the threshold value.

Values: 100us, 300us, 1ms , 3ms , 10ms , 30ms , 100ms, 300ms, 1s

Units: N/A

Example of Use: sp.slowtimeconst = ”10ms”

sp.threshold

Description: Threshold specifies the level difference between the outputs of the fast and slow integrators that will flag an RFI transient. This value is specified in units of volts. The maximum value is 10 V. The optimum value must be set experimentally since it depends on the Bandwidth, A/D Input Level, Fast Time Constant, and interference characteristics.

Values: float

Allowed Range: $0.0 \leq$ sp.threshold ≤ 10.0

Units: volts

Example of Use: sp.threshold = 8.5

sp.iflosource

Description: The IF LO Source flag controls whether an IF drawer has its input center frequency set by its internal synthesizer with 10 kHz resolution or by the external synthesizer with 10 Hz resolution. There is only one external synthesizer that must be shared between IF drawer so the normat operation is with internal LO's. The external synthesizer is same one used for the spectral processor variable master clock, so it is not available in pulsar modes.

Values: Int, Ext

Units: N/A

Example of Use: sp.iflosource = Int

sp_stor.tapeid

Description: Tape ID is a four-digit tape identification number found on the tape cartridge.

Values: tape id string

Units: N/A

Example of Use: sp_stor.tapeid = ”0032”

sp_stor.tapedirect

Description: Tape Direct determines whether data are written directly to tape, instead of to disk.

Values: NO , YES

Units: N/A

Example of Use: sp_stor.tapedirect = NO

sp_stor.eoftrigger

Description: EOF Trigger specifies whether file marks are written to tape at scan change or after pauses of a length set by EOF Delay.

Values: ” No EOF ”, ”Scan End”, Delayed

Units: N/A

Example of Use: sp_stor.eoftrigger = ”Scan End”

sp_stor.eofdelay

Description: If EOF Trigger is set to write after a pause in tape writing, EOF Delay sets this pause length in seconds.

Values: float

Units: seconds

Example of Use: sp_stor.eofdelay = 1.5

sp_stor.eofcontrol

Description: EOF Control tells whether an end-of-file will be written to tape. -1 = no EOF, 0 = at a new scan, otherwise the value is equal to EOF Delay. This parameter is read-only.

Values: integer

Units: N/A

Example of Use: sp_stor.eofcontrol = -1

sp_stor.writefile

Description: Write File determines whether all data, no data, or only samples of data are written to disk or tape. Sample Interval sets the number of data records between writes, if this option is selected.

Values: " All Data ", " No Data ", "Sample Only"

Units: N/A

Example of Use: sp_stor.writefile = "All Data"

sp_stor.filecontrol

Description: File Control tells whether a data record will be written. 0 = no writes, otherwise a record will be written when (record# % fileControl == 0).

Values: integer

Units: N/A

Example of Use: sp_stor.filecontrol = 5

sp_stor.sampleinterval

Description: Sample Interval is the number of records between data writes when the Write File parameter is set to 'Sample'.

Values: integer

Units: N/A

Example of Use: sp_stor.sampleinterval = 2

sp_stor.filename

Description: File Name is the name of the file to which data are being written. This is determined automatically as a combination of time, date, and device ID.

Values: filename

Units: N/A

Example of Use: sp_stor.filename = ”/users/blinky/data.dat”

sp_stor.collate

Description: Collate controls whether data from spectral processor racks A and B are collated into one data record before being written to disk or tape.

Values: YES, NO

Units: N/A

Example of Use: sp_stor.collate = YES

sp_stor.log

Description: Log controls whether a log of all data records is generated.

Values: YES, NO

Units: N/A

Example of Use: sp_stor.log = YES

sp_stor.flushbuffer

Description: Flush Buffer causes unwritten records to be written to disk or tape. This is normally used only when there is a problem with the data flow such as data interruption from one of the spectral processor racks when collation is turned on.

Values: integer

Units: N/A

Example of Use: sp_stor.flushbuffer = 1

4.6.3 Spectrometer

- spm.configuration
- spm.fastsamplers
- spm.slowsamplers
- spm.numberslowsamplers
- spm.relativebandwidth
- spm.samplermode
- spm.correlations
- spm.levels

- spm.numberofifs
- spm.bandwidth
- spm.requestedintegrationtime
- spm.aswitchperiod
- spm.balance
- spm.balancemode

spm.configuration

Description: Selects the base operational mode and and quadrant assignments of the GBT Spectrometer.

Values: Testing, A1, ”A1 B1”, ”A1 B1 C1”, ”A1 B1 C1 D1”, ”A1 B1 C2”, A2, ”A2 B1”, ”A2 B1 C1”, ”A2 B2”, A4, Pulsar

Units: N/A

Example of Use: spm.configuration = ”A1”

spm.fastsamplers

Description: The 1.6 GHz samplers are individually assigned to specific banks.

Values: banka, bankb, bankc, bankd, notused

Units: N/A

Example of Use: spm.fastsamplers = banka

spm.slowsamplers

Description: The 100 MHz samplers are assigned to specific banks in groups of 8.

Values: banka, bankb, bankc, bankd, notused

Units: N/A

Example of Use: spm.slowsamplers = bankb

spm.numberslowsamplers

Description: The Number of slow samplers used for each bank of the spectrometer. Valid selections are 1, 2, 4, or 8.

Values: 1, 2, 4, 8

Units: N/A

Example of Use: spm.numberslowsamplers = 2

spm.relativebandwidth

Description: 1.6 GHz samplers: 200 vs. 800 MHz bandwidth; or 100 MHz samplers:12.5 vs. 50 MHz bandwidth.

Values: narrow, wide

Units: N/A

Example of Use: spm.relativebandwidth = wide

spm.samplermode

Description: Selects display of Fast or Slow sampler information.

Values: Fast, Slow

Units: N/A

Example of Use: spm.samplermode = Fast

spm.correlations

Description: Selects auto-correlation or cross-correlation. of signals

Values: auto, cross

Units: N/A

Example of Use: spm.correlations = auto

spm.levels

Description: Selects 3-level or 9-level A/D sampling.

Values: 3, 9

Units: N/A

Example of Use: spm.levels = 9

spm.numberofifs

Description: Selects the desired total number of IFs (IF frequencies times number of polarizations) for the GBT Spectrometer setup.

Values: 1, 2, 4, 8, 16, 32

Units: N/A

Example of Use: spm.numberofifs = 4

spm.bandwidth

Description: Selects the desired bandwidth for the GBT Spectrometer setup.

Values: 800MHz, 200MHz, 50MHz, 12.5MHz

Units: N/A

Example of Use: spm.bandwidth = ”200MHz”

spm.requestedintegrationtime

Description: Controls the approximate time in seconds (within 2 microseconds) of one data record.

Values: float

Units: seconds

Example of Use: spm.requestedintegrationtime = 10.0

spm.aswitchperiod

Description: The Switch Period specifies the time in seconds of a full switch cycle being used in the spectrometer.

Values: float

Units: seconds

Example of Use: spm.switchperiod = 1.0

spm.balance

Description: Turns the balancing of the input to the samplers On or Off

Values: Off, On

Units: N/A

Example of Use: spm.balance = Off

spm.balancemode

Description: Controls whether balancing is done upon activation or prior to a scan

Values: ScanStart, Immediate

Units: N/A

Example of Use: spm.balancemode = ScanStart

4.6.4 Berkley-CalTech Pulsar Machine

- bcpm.operatingmode
- bcpm.submanagersused
- bcpm.sumpolarizations
- bcpm.calusedflag
- bcpm.setpower
- bcpm.datastorage
- bcpm.voltageregister
- bcpm.sampletime
- bcpm.samplechoice
- bcpm.timingtype
- bcpm.channelbandwidth
- bcpm.setif
- bcpm.numberphasebins
- bcpm.period
- bcpm.dispersionmeasure

- bcpm.filesize
- bcpm.centerfrequency
- bcpm.targetname
- bcpm.basename

bcpm.operatingmode

Description: This selects the mode of operation of the BCPM.

Values: search, voltage_sampling, timing, monitor

Units: N/A

Example of Use: bcpm.operatingmode = monitor

bcpm.submanagersused

Description: This selects the used of submanagers of the BCPM.

Values: BCPM1, BCPM2, "BCPM1 & BCPM2"

Units: N/A

Example of Use: bcpm.submanagersused = BCPM2

bcpm.sumpolarizations

Description: This flag indicates if the polarizations should be summed in the BCPM

Values: YES, NO, Not_Used

Units: N/A

Example of Use: bcpm.sumpolarizations = YES

bcpm.calusedflag

Description: This flag indicates if the used_flag should be calmed in the BCPM

Values: YES, NO

Units: N/A

Example of Use: bcpm.calusedflag = NO

bcpm.setpower

Description: This parameter iniciates the setpower function of the BCPM when changed from a zero to non-zero integer

Values: integer

Units: N/A

Example of Use: bcpm.setpower = 1

bcpm.datastorage

Description: This flag indicates the media type for data storage of BCPM generatedpulsar data.

Values: Disk, Tape

Units: N/A

Example of Use: bcpm.datastorage = Tape

bcpm.voltageregister

Description: This flag indicates the register from which to sample voltages in the voltage sampling mode of the BCPM

Values: POLA0, POLA1, POLA2, POLA3, POLB4, POLB5, POLB6, POLB7, Not_Used

Units: N/A

Example of Use: bcpm.voltageregister = POLA0

bcpm.sampletime

Description: Reduction time for reducing the time and/or frequency resolution in the BCPM

Values: X1, X2, X4, X8

Units: N/A

Example of Use: bcpm.sampletime = X2

bcpm.samplechoice

Description: Reduction factor for reducing the time and/or frequency resolution in the BCPM

Values: ”Total Power”, Voltage, None

Units: N/A

Example of Use: bcpm.samplechoice = ”Total Power”

bcpm.timingtype

Description: Which sub-mode of BCPM timing mode is to be used.

Values: cal, demo, pulsar

Units: N/A

Example of Use: bcpm.timingtype = pulsar

bcpm.channelbandwidth

Description: This parameter is the bandwidth for the channels used in BCPM1 andBCPM2 respectively.

Values: ”1.74 MHz”, ”1.40 MHz”, ”1.00 MHz”, ”0.70 MHz”, ”0.50 MHz”, ”0.35 MHz”, ”0.25 MHz”

Units: N/A

Example of Use: bcpm.channelbandwidth = ”1.40 MHz”

bcpm.setif

Description: This parameter sets the input if for the channels used in BCPM1 andBCPM2 respectively.

Values: 1, 2, 3

Units: N/A

Example of Use: bcpm.setif = 1

bcpm.numberphasebins

Description: Number of bins across the pulsar's pulse period.

Values: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192

Units: N/A

Example of Use: bcpm.numberphasebins = 1024

bcpm.period

Description: Period of the pulsar (milliseconds) under observation using the BCPM.

Values: float

Units: milliseconds

Example of Use: bcpm.period = 0.234567

bcpm.dispersionmeasure

Description: Dispersion Measure of the pulsar under observation using the BCPM.

Values: float

Units: parsecs/cm$^-3$

Example of Use: bcpm.dispersionmeasure = 876.34

bcpm.filesize

Description: Sky File size.

Values: integer

Units: seconds

Example of Use: bcpm.filesize = 1000

bcpm.centerfrequency

Description: Center frequency of IF going into BCPM.

Values: float

Units: MHz

Example of Use: bcpm.centerfrequency = 630.0

bcpm.targetname

Description: Right Ascension

Values: any string

Units: N/A

Example of Use: bcpm.targetname = ”0329+54”

bcpm.basename

Description: Right Ascension

Values: any string

Units: N/A

Example of Use: bcpm.basename = ”0329+54”