Chapter 1

NRAO

NRAO-related modules and tools

include "nrao.g"

Description
Package for NRAO-related functionality

1.1 gbt - Module

Module for GBT data processing
include "gbt.g"

Description
Various functions and tools for processing GBT logs and data. Most can be invoked either from within Glish or from the UNIX command line. Both forms are documented here.
Functions

gbtlogfiller	Join GBT logs into a single AIPS++Table
gbtlogview	View GBT logs

Tools

dcr	A tool for analyzing DCR data from the GBT.
gbtmsfiller	A tool for filling GBT data to an aips++ MeasurementSet.

1.1.1 dcr - Tool

Package nrao

Module gbt

A tool for analyzing DCR data from the GBT.
include "dcr.g"

Constructors

dcr	Construct dcr tool

Functions

baselinefit	Fit a baseline
done	Close and exit the DCR tool
focus	Fit a parabola to a focus observation
focusScan	Reduce a single focus scan.
gauss	Fit a gaussian function to the data.
getGO	Return the GO keywords
get_tant	Returns Antenna Temperature for a given scan
getscan	Return data for the given scan
guessmode	Guess the switching signal mode
listscans	List scan numbers
plot_RA_Dec	Plot Dec vs. RA
plot_dap_time	Plot a data associated parameter (DAP) vs. time
plot_focus_time	Plot focus against time
plot_gain_time	Plot relative gain (normalized counts/K) vs. time - not yet implemented
plot_phase_dec	Plot data counts vs. Dec for a given phase
plot_phase_ra	Plot data counts vs. RA for a given phase
plot_phase_time	Plot data counts vs. time for a given phase
plot_sidelobe	Plot sidelobes
plot_tant_Dec	Plot Antenna Temperature vs. Dec
plot_tant_RA	Plot Antenna Temperature vs. RA
plot_tant_time	Plot Antenna Temperature vs. time
plot_tsrc_time	Plot Tsrc against time
plotscans	Plot several scans contiguously
point1	Reduce a 1 scan pointing observation
point2	Reduce a 2 scan pointing observation
point4	Reduce a 4 scan pointing observation
scanSummary	Print a summary of the scans in the GUI message window
test_srp	Test whether a specified scan, receiver, phase exists
tip	Reduce a tipping scan.
tsys	Calculate the system temperature
type	Return the type of this tool.

Description
This `tool` is an in-house suite of functions intended to be used exclusively with digital continuum receiver (DCR) data from the Green Bank Telescope (GBT). The `tool` consists of 2 GUI components: A panel of buttons which offers an interface to some of the common functionality, and a plotter window which displays results when applicable. The complete functionality of the dcr tool is available from the command line interface. The tool is aimed at satisfying the requirements for commissioning the GBT, primarily, but it may find use as a general purpose continuum analysis tool as well.

dcr - Constructor

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Construct dcr tool

Synopsis
dcr(filename)

Description
This constructs the `tool` which provides numerous functions for analyzing DCR data. The constructor requires the name of a data file which must be an AIPS++ Measurement Set. To access data from a different data file it is necessary to open a new instance of the tool.

Arguments

filename	filename of an AIPS++ Measurement Set which contains the data Allowed:string Default: unset

Returns
Tool

---

dcr.getscan - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Return data for the given scan

Synopsis
getscan(scan, getFocus)

Description
The most fundamental routine in the dcr tool. This returns a record containing information about the selected scan, including header info and the data. It also sets up variables which are used internal to the dcr tool by other routines. The focus values are not included by default in order to improve efficiency.

Arguments

scan	Scan number Allowed:Int
getFocus	flag to determine whether focus values will be retrieved Allowed:Boolean Default: F

Returns
record

---

dcr.listscans - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

List scan numbers

Synopsis
listscans()

Description
Returns an array containing the scan numbers.

Returns
integer array

---

dcr.done - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Close and exit the DCR tool

Synopsis
done()

Description
Closes the DCR tool and removes it from memory.

Returns
boolean

---

dcr.getGO - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Return the GO keywords

Synopsis
getGO(scan)

Description
The astronomer's interface to controlling the GBT (the GO program) produces FITS files containing keywords relevant to the observing setup. This function returns a record containing those keywords.

Arguments

scan	Scan number Allowed:Int

Returns
record

---

dcr.guessmode - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Guess the switching signal mode

Synopsis
guessmode(scan)

Description
By looking at the number and arrangement of phases, this routine makes a guess at the switching signal mode. It has been made redundant by an entry in the GO header which conveys the same parameter directly.

Arguments

scan	Scan number Allowed:Int

Returns
string

---

dcr.baselinefit - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Fit a baseline

Synopsis
baselinefit(xarray, yarray, ord, range, plotflag)

Description
This function provides a baseline fit to the given data. The residuals are returned.

Arguments

xarray	x values Allowed:double
yarray	y values Allowed:double
ord	Order of the polynomial Allowed:int
range	Channel range to be fit Allowed:array
plotflag	Plot the fit? Allowed:boolean Default: T

Returns
vector of residuals

---

dcr.focus - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Fit a parabola to a focus observation

Synopsis
focus(filename)

Arguments

filename	name of ascii file on disk Allowed:string

Returns
boolean

Description
Read an ascii file containing focus values from a sequence of point/focus combinations, and fit a parbola to find the peak value on the curve.

Example

```
include 'dcr.g'
md := dcr('GBTdata.ms')  # suppose we have this data set on disk
md.focus('focus.data')
```

where in this case the file focus.data might have this form:

```
1.2   15.5
1.3   15.7
1.4   16.0
1.5   16.1
1.6   15.9
1.7   15.6
1.8   15.4
```

---

dcr.get_tant - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Returns Antenna Temperature for a given scan

Synopsis
get_tant(scan, receiver, cal_value)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int Default: 1
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1

Returns
boolean

Description
Returns an array of antenna temperatures.

---

dcr.gauss - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Fit a gaussian function to the data.

Synopsis
gauss(xarray, yarray, height, width, center, plotflag)

Arguments

xarray	X values Allowed:double
yarray	Y values Allowed:double
height	initial guess for height Allowed:float
width	initial guess for width Allowed:float
center	initial guess for center Allowed:float
plotflag	plot the result? Allowed:boolean Default: T

Returns
vector of residuals.

Description
This function fits a gaussian to the data. The residuals are returned.

---

dcr.plot_dap_time - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot a data associated parameter (DAP) vs. time

Synopsis
plot_dap_time(scan, colName)

Arguments

scan	Scan number Allowed:Int
colName	colName from the DAP subtable Allowed:string

Returns
boolean

---

dcr.plot_gain_time - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot relative gain (normalized counts/K) vs. time - not yet implemented

Synopsis
plot_gain_time(scan, receiver)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int

Returns
boolean

---

dcr.plot_RA_Dec - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot Dec vs. RA

Synopsis
plot_RA_Dec(scan)

Arguments

scan	Scan number Allowed:Int

Returns
boolean

---

dcr.plot_phase_ra - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot data counts vs. RA for a given phase

Synopsis
plot_phase_ra(scan, receiver, phase)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
phase	Phase number Allowed:Int

Returns
boolean

---

dcr.plot_phase_dec - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot data counts vs. Dec for a given phase

Synopsis
plot_phase_dec(scan, receiver, phase)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
phase	Phase number Allowed:Int

Returns
boolean

---

dcr.plot_phase_time - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot data counts vs. time for a given phase

Synopsis
plot_phase_time(scan, receiver, phase)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
phase	Phase number Allowed:Int

Returns
boolean

---

dcr.plot_tsrc_time - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot Tsrc against time

Synopsis
plot_tsrc_time(scan, receiver, cal_value)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1

Returns
boolean

Description
Plot source temperature against time. This only works when the switching phases are set to switched power with cal.

---

dcr.plot_focus_time - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot focus against time

Synopsis
plot_focus_time(scan, param)

Arguments

scan	Scan number Allowed:Int
param	Focus parameter Allowed:string Default: SR_XP

Returns
boolean

Description
Plots a focus value against time. The focus parameter can be any of SR_XP, SR_YP, SR_ZP, SR_XT, SR_YT, or SR_ZT for gregorian focus observing, or ANTPOSPF, PF_FOCUS, or PF_ROTATION for prime focus observing.

---

dcr.focusScan - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Reduce a single focus scan.

Synopsis
focusScan(scan, receiver, cal_value, param, order, archive)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int Default: 0
cal_value	Cal value Allowed:float Default: 1
param	Focus parameter Allowed:string Default: SR_XP
order	Order of polynomial to fit Allowed:int Default: 2
archive	Archive the result? Allowed:boolean Default: F

Returns
boolean

Description
Plots a focus value against time. The focus parameter can be any of SR_XP, SR_YP, SR_ZP, SR_XT, SR_YT, or SR_ZT for gregorian focus observing, or PF_X, PF_FOCUS, or PF_ROTATION for prime focus observing. A polynomial is fit the to data, and the peak value from the fit is reported.

---

dcr.plot_tant_time - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot Antenna Temperature vs. time

Synopsis

plot_tant_time(scan, receiver, cal_value)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int Default: 1
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1

Returns
boolean

---

dcr.plot_tant_RA - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot Antenna Temperature vs. RA

Synopsis
plot_tant_RA(scan, receiver, cal_value)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int Default: 1
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1

Returns
boolean

---

dcr.plot_tant_Dec - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot Antenna Temperature vs. Dec

Synopsis
plot_tant_Dec(scan, receiver, cal_value)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int Default: 1
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1

Returns
boolean

---

dcr.plot_sidelobe - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot sidelobes

Synopsis
plot_sidelobe(scan, receiver, basepct, bottom)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int Default: 1
basepct	Percent of scan to use for baseline Allowed:Int Default: 10
bottom	dB level to use for cutoff at bottom of plot Allowed:Int Default: -70

Description
Plots the sidelobes on the traditional dB scale. Best if used on a scan intended for this purpose.

Returns
boolean

---

dcr.plotscans - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Plot several scans contiguously

Synopsis
plotscans(bscan, escan, receiver, phase)

Arguments

bscan	beginning scan number Allowed:Int
escan	ending scan number Allowed:Int
receiver	Receiver number Allowed:Int
phase	Phase number Allowed:Int

Returns
boolean

---

dcr.point1 - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Reduce a 1 scan pointing observation

Synopsis
point1(scan, receiver, xaxis, cal_value, basepct, plotflag)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
xaxis	1=RA, 2=dec Allowed:Int
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1
basepct	percentage of scan to use on the edges for a baseline fit Allowed:Int Default: 10
plotflag	flag for plotting the data Allowed:Boolean Default: T

Description
Reduces a pointing observation made with the GBT. Baselines are fit from the two edges of the scan only. The parameter basepct specifies the percentage of the total scan used from each edge. For example, a basepct of 10 means that a tenth of the scan at each of the two edges is used in calculating the baseline. If xaxis=0, the function will attempt to determine whether RA or Dec is appropriate for the given scan.

Returns
boolean

---

dcr.point2 - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Reduce a 2 scan pointing observation

Synopsis
point2(scan, receiver, cal_value)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1

Description
Automated reduction of 2-scan pointing observations with the GBT.

Returns
boolean

---

dcr.point4 - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Reduce a 4 scan pointing observation

Synopsis
point4(scan, receiver, cal_value, plotflag)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
cal_value	The calibration value for deriving counts/K Allowed:float Default: 1
plotflag	flag for plotting the data Allowed:Boolean Default: T

Description
Automated reduction of 4-scan pointing observations with the GBT.

Returns
boolean

---

dcr.test_srp - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Test whether a specified scan, receiver, phase exists

Synopsis
test_srp(scan, receiver, phase)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int
phase	Receiver number Allowed:Int

Description
This is used to test whether a given combination of scan, receiver, and phase

exist in the opened MeasurementSet. This returns T(rue) if the combination does NOT exist.

Returns
boolean

---

dcr.scanSummary - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Print a summary of the scans in the GUI message window

Synopsis
scanSummary()

Returns
boolean

---

dcr.tsys - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Calculate the system temperature

Synopsis
tsys(scan, receiver)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int

Returns
boolean

---

dcr.tip - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Reduce a tipping scan.

Synopsis
tip(scan, receiver)

Arguments

scan	Scan number Allowed:Int
receiver	Receiver number Allowed:Int

Returns
boolean

Description
Reduce a tipping scan. Uses least-squares to derive an opacity.

---

dcr.type - Function

Package	**nrao**
Module	**gbt**
Tool	**dcr**

Return the type of this tool.

Synopsis
type()

Returns
string

Description
This is required for the toolmanager to manage this tool. This function always returns "dcr".

1.1.2 gbtmsfiller - Tool

Package nrao
Module gbt

A tool for filling GBT data to an aips++ MeasurementSet.
include "gbtmsfiller.g"

Constructors

gbtmsfiller	Construct gbtmsfiller tool

Functions

backend	Get the current backend type
compresscalcols	Get the value of the compresscalcols flag.
dcbias	Get the specific value of the dcbias to use during the van Vleck correction.
done	Stop filling and close the tool.
fillall	Start filling all of the scans in the scan log
filllags	Get the value of the filllags flag.
fillnext	Start filling the next scan in the scan log
fillrawfocus	Get the value of the fillrawfocus flag.
fillrawpointing	Get the value of the fillrawpointing flag.
fixlags	Get the fixlags toggle value.
isattached	Does the filler have a valid project directory
maxscan	Get the current maximum scan number for data to be filled
maxtime	Get the current maximum time for data to be filled
minbiasfactor	Get the minbiasfactor value.
minscan	Get the current minimum scan number for data to be filled
mintime	Get the current minimum time for data to be filled
more	Are there more scans to process in the scan log?
msdirectory	Get the current directory of all output MSs
msrootname	Get the current msrootname
newms	Force a new MS for all currently filling backends.
object	Get the current object string to match
oneacsms	Get the value of the oneacsms flag.

project	Get the current project directory
setbackend	Limit the backends to fill to one type
setcompresscalcols	Set the flag which controls whether the calibration columns are compressed.
setdcbias	Set a specific bias to use during the van Vleck correction.
setfilllags	Set the flag which controlls whether the LAG_DATA data column is filled.
setfillrawfocus	Set the flag which controlls whether the raw focus values are saved (filled).
setfillrawpointing	Set the flag which controlls whether the raw pointings are saved (filled).
setfixlags	Toggle whether bad ACS lags should be fixed, if possible.
setmaxscan	Change the maximum scan number for data to be filled
setmaxtime	Change the maximum time for data to be filled
setminbiasfactor	Attempt to correct for the truncation error in the ACS samplers.
setminscan	Change the minimum scan number for data to be filled
setmintime	Change the minimum time for data to be filled
setmsdirectory	Change the directory of all output MSs
setmsrootname	Change the root name of all output MSs
setobject	Limit the fill to objects matching this string
setoneacsms	Set the flag which controlls whether a single ACS MeasurementSet is filled.
setproject	Set the project directory
setsmooth	Choose the type of smoothing for spectrometer data.
setusebias	Set the flag which controlls whether the bias is calculated during the van Vleck correction.
setusehighcal	Set the flag which controlls which CAL_TEMP is used from the receiver cal table.
setvv	Choose the type of vanVleck correction for spectrometer data.
smooth	Get the value of type of smoothing being used.
status	Report the status of the filler
type	Return the type of this tool.
update	Reopen and resync with the scan log
usebias	Get the value of the usebias flag.
usehighcal	Get the value of the usehighcal flag.
vv	Get the value of type of vanVleck correction being used.

Description

This `tool` is intended to be run by the GBT operators during normal operation of the telescope. Several MeasurementSets will typically be filled at the same time (one for each backend in use by each project during an observing session). The filling process is driven by the ScanLog.fits file for each project. This tool is under development and the exact operational details have not yet been finalized. It is anticipated that most interations with this `tool` will be done through its GUI.

Example

No example available yet.

This space reserved for comments about the example, when it becomes available.

gbtmsfiller - Constructor

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Construct gbtmsfiller tool

Synopsis
gbtmsfiller()

Returns
object

Description
This creates the gbtmsfiller `tool`. The resulting `tool` is not yet ready to fill. The project directory must first be specified.

Example

```
gbtfiller := gbtmsfiller()
gbtfiller.setproject('/home/gbtdata/AGBT02A_028_01')
# the filler is now ready to fill
gbtfiller.fillall()
```

The plan is that once a project is specified, subsequent filling will happen automatically when new data arrives. For the time being, the decision to fill to a new MS as opposed to continuing to append to the old MS will be left up to the operator. It is expected that reasonable heuristics will be developed to make this process automatic as well although the ability to force a new MS will continue to be available.

---

gbtmsfiller.isattached - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Does the filler have a valid project directory

Synopsis
isattached()

Returns
boolean

Description
Does the current project directory exist and is there a ScanLog.fits file there? When this function returns True, the filler is attached to a project and is ready to begin filling.

---

gbtmsfiller.fillall - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Start filling all of the scans in the scan log

Synopsis
fillall(async)

Arguments

async	Run this command asyncronously Allowed:boolean Default: F

Returns
boolean

Description
Start filling. All of the known data on disk at the time of the last update of the scan log is filled. When run asyncronously, this function returns before the filling has completed. Watch the log in that case to see the progress of the filling.

---

gbtmsfiller.fillnext - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Start filling the next scan in the scan log

Synopsis
fillnext(async)

Arguments

async	Run this command asyncronously Allowed:boolean Default: F

Returns
boolean

Description
Start filling. The next scan in the scan log is filled. It is anticipated that this function will be automatically called during normal operation whenever new data is written to disk (update is called first). When run asyncronously, this function returns before the filling has completed. Watch the log in that case to see the progress of the filling.

---

gbtmsfiller.update - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Reopen and resync with the scan log

Synopsis
update(async)

Arguments

async	Run this command asyncronously Allowed:boolean Default: F

Returns
boolean

Description
This should be called whenever the scan log FITS file has changed so that the filler can know about all of the data on disk. When run asyncronously, this function returns before the update has completed. Watch the log in that case to see the progress of the update.

---

gbtmsfiller.more - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Are there more scans to process in the scan log?

Synopsis
more()

Returns
boolean

Description
The answer to this question reflects the state of the scan log at the last update. If the scan log has since changed on disk, update must first be called in order that the filler can see those changes.

---

gbtmsfiller.status - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Report the status of the filler

Synopsis
status()

Returns
record

Description
Returns the status record.

status - Record	the status record
status.attached	is the filler attached Allowed: boolean
status.nextscan	pending scan number to be processed Allowed: integer
status.nextdmjd	pending DMJD value to be processed Allowed: double
status.nexttimestamp	pending scan timestamp to be processed Allowed: double
status.project	project being filled Allowed: string
status.msdirectory	The directory where any MSs are being written Allowed: string
status.object	object constraint Allowed: string
status.mintime	minimum time constraint Allowed: string
status.maxtime	maximum time constraint Allowed: string
status.minscan	minimum scan number constraint Allowed: integer
status.maxscan	maximum scan number constraint Allowed: integer
status.fillrawpointing	Are the raw pointings being saved? Allowed: boolean
status.fillrawfocus	Are the raw focus data being saved? Allowed: boolean
status.filllags	Are the raw lags being filled? Allowed: boolean
status.vv	The current vanVleck correction Allowed: string
status.smooth	The current smoothing function

	Allowed: string
status.usehighcal	Is the $HIGH_CAL$ being used? (old data only)
	Allowed: boolean
status.compresscalcols	Are the spectral line calibration columns compressed?
	Allowed: boolean
status.usebias	Is the bias being calculated for the ACS vanVleck correction?
	Allowed: boolean
status.backendtype	backend type constraint
	Allowed: string
status.oneacsms	Is a single ACS MS being filled?
	Allowed: boolean
status.dcbias	Current DC bias value.
	Allowed: double
status.minbiasfactor	Minimum bias factor.
	Allowed: integer
status.fixlags	Try and fix the ACS lags.
	Allowed: boolean
status.dcr	Status of the MS being filled with DCR data
	Allowed: backend status record
status.holo	Status of the MS being filled with Holography data
	Allowed: backend status record
status.sp	Status of the MS being filled with SpectralProcessor data
	Allowed: backend status record
status.acs	Record of status records for ACS (spectrometer) data
	Allowed: record

The ACS (spectrometer) status record has the following structure:

acs - Record	Record of status records for ACS data
acs.ABCD	Status of the MS being filled with all ACS data (banks A,B,C,D)
	Allowed: backend status record
acs.A	Status of the MS being filled with bank A ACS data
	Allowed: backend status record
acs.B	Status of the MS being filled with bank B ACS data
	Allowed: backend status record
acs.C	Status of the MS being filled with bank C ACS data
	Allowed: backend status record
acs.D	Status of the MS being filled with bank D ACS data
	Allowed: backend status record

Each backend has a status record (the ACS (spectrometer) backend has 5 status records, one for the single MS being filled with data from all banks and one for each possible bank, all grouped in the acs record). The backend status record looks like this:

backend status - Record	the backend status record
backend status.ms	The name of the MS being filled. This is unset if it is not being filled. Allowed: string
backend status.nrows	The total number of rows in that MS. This is -1 if it is not being filled. Allowed: integer

If the attached field is False, all of the other fields may not reflect the true state of the filler. The project, object, mintime, maxtime, minscan, maxscan, fillrawpointing, fillrawfocus, filllags, vv, smooth, usehighcal, compresscalcols, usebias, oneacsms and backendtype constraints can all be set via this DO interface. When the filler is at the end of the scan log (more() return False) then nextscan will have a value of -1 and nextdmjd and nextimestamp will have values of 0.0.

---

gbtmsfiller.setproject - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the project directory

Synopsis
setproject(project)

Arguments

project	The project directory Allowed:string

Returns
boolean

Description
Setting the project directory causes any currently open ScanLog.fits file as well as any MeasurementSets currently being filled to be closed. The next fill will open the new ScanLog.fits and begin filling information found there.

---

gbtmsfiller.project - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current project directory

Synopsis
project()

Returns
string

Description
Get the current project directory known to the filler. This returns an empty string if no project has been set.

---

gbtmsfiller.setbackend - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Limit the backends to fill to one type

Synopsis
setbackend(backend)

Arguments

backend	The backend type to fill Allowed:string

Returns
boolean

Description

Possible types are ANY, HOLO, DCR, SP, and ACS. SPECTRALPROCESSOR is synonymous with SP and SPECTROMETER is synonymous with ACS. Any other types from that specified here will be ignored. The default, ANY, implies that all of the known types will be filled. This argument is case insensitive.

---

gbtmsfiller.backend - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current backend type

Synopsis
backend()

Returns
string

Description
Get the current backend type being filled. The default is ANY, implying that any backends known to the filler are filled.

---

gbtmsfiller.setmsdirectory - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Change the directory of all output MSs

Synopsis
setmsdirectory(msdirectory)

Arguments

msdirectory	The directory of all output MSs Allowed:string

Returns
boolean

Description
The output MSs are all placed in this directory. This function allows you to change the default, which is the current directory.

---

gbtmsfiller.msdirectory - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current directory of all output MSs

Synopsis
msdirectory()

Returns
string

---

gbtmsfiller.setmsrootname - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Change the root name of all output MSs

Synopsis
setmsrootname(msrootname)

Arguments

msrootname	The root name of all output MS Allowed:string

Returns
boolean

Description
The output MSs are named using this string plus the backend type plus a version number: msrootname_backendtype. This function allows you to change the default, which is the base name of the project directory. The backendtype is one of HOLO, DCR, SP, or ACS. If multiple ACS MeasurementSets are being filled. There will be an additional _A, _B, _C, or _D appended to the name as necessary.

---

gbtmsfiller.msrootname - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current msrootname

Synopsis
msrootname()

Returns
string

---

gbtmsfiller.setmintime - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Change the minimum time for data to be filled

Synopsis
setmintime(mintime)

Arguments

mintime	The minimum time. Allowed:string

Returns
boolean

Description
Only data taken after mintime will be filled. The default is for there to be no minimum time and fill all data. If a sub-scan spans mintime (i.e. starts before mintime but ends after mintime) the entire sub-scan will be filled. The time string is specified using the FITS time convention:
YYYY-MM-DDTHH:MM:SS.sss.

---

gbtmsfiller.mintime - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current minimum time for data to be filled

Synopsis
mintime()

Returns
string

---

gbtmsfiller.setmaxtime - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Change the maximum time for data to be filled

Synopsis

setmaxtime(mintime)

Arguments

mintime	The maximum time. Allowed:string

Returns
boolean

Description
Only data taken before maxtime will be filled. The default is for there to be no maximum time and fill all data. If a sub-scan spans maxtime (i.e. starts before maxtime but ends after maxtime) the entire sub-scan will be filled. The time string is specified using the FITS time convention:
YYYY-MM-DDTHH:MM:SS.sss.

---

gbtmsfiller.maxtime - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current maximum time for data to be filled

Synopsis
maxtime()

Returns
string

---

gbtmsfiller.setminscan - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Change the minimum scan number for data to be filled

Synopsis
setminscan(minscan)

Arguments

minscan	The minimum scan number. Allowed:integer

Returns
boolean

Description
Only data having a scan number greater than or equal to minscan will be filled. Specifying a value of -1 (the default at startup) indicates that there is no minimum scan number.

---

gbtmsfiller.minscan - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current minimum scan number for data to be filled

Synopsis
minscan()

Returns
integer

---

gbtmsfiller.setmaxscan - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Change the maximum scan number for data to be filled

Synopsis
setmaxscan(minscan)

Arguments

minscan	The maximum scan number. Allowed:integer

Returns
boolean

Description
Only data having a scan number less than or equal to maxscan will be filled. Specifying a value of -1 (the default at startup) indicates that there is no maximum scan number.

---

gbtmsfiller.maxscan - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current maximum scan number for data to be filled

Synopsis
maxscan()

Returns
integer

---

gbtmsfiller.setobject - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Limit the fill to objects matching this string

Synopsis
setobject(object)

Arguments

object	The object string to match Allowed:string	

Returns
boolean

Description
Only scans with objects matching this string will be filled. Simple wild cards involving * are allowed. If this string is just that wildcard (the default when this `tool` is created) then all objects will match.

---

gbtmsfiller.object - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the current object string to match

Synopsis
object()

Returns
string

---

gbtmsfiller.setfillrawpointing - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the flag which controlls whether the raw pointings are saved (filled).

Synopsis
setfillrawpointing(fillrawpointing)

Arguments

fillrawpointing	When true, raw pointings are filled. Allowed:boolean Default: F

Returns
boolean

Description
The GBT Antenna FITS file contains information on the telescope pointings. The pointing information is normally sampled 10 times a second. The filler computes a mean pointing direction over each integration and stores that value in the DIRECTION column of the POINTING subtable. When the fillrawpointing flag is F(alse) (the default value) the raw, unaveraged, pointings are not saved in the MeasurementSet in order to save space. This behavior can be changed by setting this flag to T. When this flag is true, the raw pointings can be found in the NRAO_GBT_POINTING subtable. This decision must be made before the MeasurementSet is filled.

---

gbtmsfiller.fillrawpointing - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of the fillrawpointing flag.

Synopsis
fillrawpointing()

Returns
boolean

---

gbtmsfiller.setfillrawfocus - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the flag which controlls whether the raw focus values are saved (filled).

Synopsis
setfillrawfocus(fillrawfocus)

Arguments

fillrawfocus	When true, raw focus values are filled. Allowed:boolean Default: F

Returns
boolean

Description
The GBT Antenna FITS file contains information on the telescope focus parameters. The focus information is normally sampled 10 times a second. The filler computes the mean focus parameters over each integration and stores that value in the NRAO_GBT_MEAN_FOCUS subtable (future versions of the filler will likely associate this information more closely with the FEED subtable). When the fillrawfocus flag is F(alse) (the default value) the raw, unaveraged, focus values are not saved in the MeasurementSet in order to save space. This behavior can be changed by setting this flag to T. When this flag is true, the raw focus values can be found in the NRAO_GBT_FOCUS subtable. This decision must be made before the MeasurementSet is filled.

---

gbtmsfiller.fillrawfocus - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of the fillrawfocus flag.

Synopsis

fillrawfocus()

Returns
boolean

---

gbtmsfiller.setfilllags - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the flag which controlls whether the LAG_DATA data column is filled.

Synopsis
setfilllags(filllags)

Arguments

filllags	When true, the LAG_DATA column is filled. Allowed:boolean Default: F

Returns
boolean

Description
This flag is only used to control filling of data from the ACS (spectrometer). Normally, when this flag is false, the raw lag data are only used to generate the FLOAT_DATA column. When this flag is true, the raw data are also filled to the LAG_DATA column. The FLOAT_DATA column will continue to be filled in that case.

---

gbtmsfiller.filllags - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of the filllags flag.

Synopsis
filllags()

Returns
boolean

---

gbtmsfiller.setvv - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Choose the type of vanVleck correction for spectrometer data.

Synopsis
setvv(vv)

Arguments

vv	Valid types are “schwab”, “old”, “none”, and “default” Allowed:string

Returns
boolean

Description
The type of vanVleck correction is only relevent to data from the spectrometer. The “schwab” correction is an implementation of a general solution developed by Fred Schwab. The “old” correction is the original vanVleck correction based on code from Arecibo. If “none” is selected, no vanVleck correction will be done. The current default, if this is not set, is “Schwab”. If the vv argument value is unrecognized, the return value will be F (false) and the default type will be used. This argument is case insensitive.

---

gbtmsfiller.vv - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of type of vanVleck correction being used.

Synopsis
vv()

Returns
string

---

gbtmsfiller.setsmooth - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Choose the type of smoothing for spectrometer data.

Synopsis
setsmooth(vv)

Arguments

vv	Valid types are “hanning”, “hamming”, “none”, and “default” Allowed:string

Returns
boolean

Description
The type of smoothing is only relevent to data from the spectrometer. This smoothing is applied after the vanVleck correction is done and before the data are fourier transformed to the spectral domain. The current default, if this is

not set, is "hanning". If the smooth argument value is unrecognized, the return value will be F (false) and the default type will be used. This argument is case insensisitive.

---

gbtmsfiller.smooth - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of type of smoothing being used.

Synopsis
smooth()

Returns
string

---

gbtmsfiller.setusehighcal - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the flag which controlls which CAL_TEMP is used from the receiver cal table.

Synopsis
setusehighcal(usehighcal)

Arguments

usehighcal	When true, HIGH_CAL_TEMP is used, else LOW_CAL_TEMP. Allowed:boolean Default: F

Returns
boolean

Description
The receiver cal tables contain measured TCAL values for a HIGH and LOW cal as a function of frequency for each receptor. In early GBT data, the IF FITS file lacked the information to determine which cal was appropriate. This flag is used to signal, for that old data, which cal value to use. It is ignored for current data since the IF fits file fully specifies which TCAL value is used. When this flag is true and the IF FITS file is missing this information, the HIGH cal value is used.

---

gbtmsfiller.usehighcal - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of the usehighcal flag.

Synopsis
usehighcal()

Returns
boolean

---

gbtmsfiller.setusebias - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the flag which controlls whether the bias is calculated during the van Vleck correction.

Synopsis
setusebias(usebias)

Arguments

usebias	When true, the bias is part of the vanVleck correction for ACS data. Allowed:boolean Default: F

Returns
boolean

Description
There is often a non-zero bias in the ACS (spectrometer) sampler levels. When this flag is true, the van Vleck correction attempts to work out what this bias is by examining the last 5 percent of the lag values.

---

gbtmsfiller.usebias - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of the usebias flag.

Synopsis
usebias()

Returns
boolean

---

gbtmsfiller.setoneacsms - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the flag which controlls whether a single ACS MeasurementSet is filled.

Synopsis
setoneacsms(oneacsms)

Arguments

oneacsms	When true, a single ACS MeasurementSet is filled with data from all ACS banks. Allowed:boolean Default: T

Returns
boolean

Description
The original filler filled each ACS bank to a separate MeasurementSet. This makes bulk processing of all ACS data (from all banks) difficult so the filler was rewritten to allow it to put all ACS data, from any bank, to the same MeasurementSet. That is now the default behavior. There may be circumstances where it is desirable to separate them out into separate MeasurementSets and this flag allows you to do that.

---

gbtmsfiller.oneacsms - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of the oneacsms flag.

Synopsis
oneacsms()

Returns
boolean

---

gbtmsfiller.setcompresscalcols - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set the flag which controls whether the calibration columns are compressed.

Synopsis
setcompresscalcols(compresscalcols)

Arguments

compresscalcols	When true, the calibration columns are compressed. Allowed:boolean Default: T

Returns
boolean

Description
In order to control data bloat, it is desirable to compress the calibration columns (e.g. COMPRESSED_DATA). The compression is done independently for each row. This only applies to spectral line data (DCR and Holography data are never compressed). When this flag is true, these columns are compressed.

---

gbtmsfiller.compresscalcols - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the value of the compresscalcols flag.

Synopsis
compresscalcols()

Returns
boolean

---

gbtmsfiller.setdcbias - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Set a specific bias to use during the van Vleck correction.

Synopsis
setdcbias(dcbias)

Arguments

dcbias	This value is used as the bias. Allowed:double Default: 0.0

Returns
boolean

Description
There is often a non-zero bias in the ACS (spectrometer) sampler levels. If the bias is thought to be known based on other scans, then it can be set here to a specific value. This is the actual bias in the samplers, and not simply the observed asymptote in the raw lags. If uselag is True, then this value is ignored in favor of the derived dcbias.

---

gbtmsfiller.dcbias - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the specific value of the dcbias to use during the van Vleck correction.

Synopsis
dcbias()

Returns
double

---

gbtmsfiller.setminbiasfactor - Function

Package	**nrao**

Module	**gbt**
Tool	**gbtmsfiller**

Attempt to correct for the truncation error in the ACS samplers.

Synopsis
setminbiasfactor(minbiasfactor)

Arguments

minbiasfactor	The minimum integer to be multiplied by 0.5/65536 and subtracted from the data.. Allowed:integer Default: -1

Returns
boolean

Description
There is a truncation (instead of a rounding) as the raw lags are converted to 32-bit integers. This introduces a systematic error in the lags. When this value is non-negative (¿=0) then the nearest integer larger than this value is found which, when multiplied by 0.5/65536 and added to the last 5to just above 0.0.

---

gbtmsfiller.minbiasfactor - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the minbiasfactor value.

Synopsis
minbiasfactor()

Returns
integer

---

gbtmsfiller.newms - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Force a new MS for all currently filling backends.

Synopsis
newms()

Returns
boolean

Description
This is not yet implemented.

---

gbtmsfiller.setfixlags - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Toggle whether bad ACS lags should be fixed, if possible.

Synopsis
setfixlags(fixlags)

Arguments

fixlags	When true, bad ACS lags are fixed, if possible. Allowed:boolean Default: False

Returns
boolean

Description

The ACS some times suffers from bad 1024-lag segments where the entire segment is obviously offset and possibly scaled differently from the surrounding lags. If this toggle is true and that bad-lag segment is not the first 1024 lags, then the filler will "fix" the bad lags. This fix is done by first subtracting the median of those bad lags from the bad lags. Then the bad lags are scaled so that their standard deviation matches the standard deviation of the 512 lags on either side of the bad lags. Finally, a constant is added to the bad lags so that the median of the bad lags matches the median of the 512 lags on either side of the bad lags. If this toggle is false, the data are flagged as bad.

---

gbtmsfiller.fixlags - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Get the fixlags toggle value.

Synopsis
fixlags()

Returns
boolean

---

gbtmsfiller.done - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Stop filling and close the tool.

Synopsis
done()

Returns
boolean

Description

This permantly closes the connection between this `tool` and any data being filled. Use this only when you are done with the current `tool`. Once this function has been called it becomes useless. Any MeasurementSets and data files currently opened when this function is called will be closed.

---

gbtmsfiller.type - Function

Package	**nrao**
Module	**gbt**
Tool	**gbtmsfiller**

Return the type of this tool.

Synopsis
type()

Returns
string

Description
This is required for the toolmanager to manage this tool. This function always returns "gbtmsfiller".

1.1.3 gbtlogfiller - Function

Package	**nrao**
Module	**gbt**

Join GBT logs into a single AIPS++Table

Synopsis
gbtlogfiller()

Description
This is a stand-alone program run from the unix prompt that joins GBT log data into a single AIPS++table.

Usage: gbtlogfiller startTime endTime outTable dir1 dir2 ...
startTime and endTime are expressed as : mm/dd/yyyy,hh:mm:ss
outTable is the output name of the resulting table
dir1, dir2, ... are the names of the log directories to be examined and potentially joined into the output table.

1.1.4 gbtlogview - Function

Package nrao
Module gbt

View GBT logs

Synopsis
gbtlogview()

Description
This Glish function allows viewing of GBT logs after filling into an AIPS++ table using gbtlogfiller.
This can be run as a stand-alone program by typing 'gbtlogview2' or 'gbtlogview3' from the command line for the 'new' or 'test' versions of the program.
The program can also be run as a Glish client; typing 'include "gbt.g"' begins execution and fires up the "GBT Log Data" GUI.
For more detailed information see Note 207.
