1.8 newsimulator - Module

Module for simulation of telescope data
include "newsimulator.g"

Description

newsimulator provides a unified interface for simulation of telescope processing. It can create a MeasurementSet from scratch or read in an existing MeasurementSet, it can predict synthesis data onto the (u,v) coordinates or single dish data onto (ra,dec) points, and it can corrupt this data through Gaussian errors or through specific errors residing in (anti-) calibration tables.
newsimulator **is substantially different from (and better than) simulator. Scripts written for one will not work for the other.**
In the observing phase, newsimulator tries to act like a (simple) telescope. You first make a newsimulator tool, with the name of the MeasurementSet that you wish to construct. Next you use the various set* methods to set up the observing (sources, spectral windows, *etc.*). Each such setup should be given a unique name that will be used in the next step. Then you call the observe method for each observing scan you wish to make. Here you specify the source name, spectral windoow name, and observing times. After this, you have a MeasurementSet that is complete but empty. In the next phase, you fill the MeasurementSet with data from a model and then corrupt the measurements (if desired). To fill it in with a model, use the predict method. Finally, to apply errors, first set up the various effects using the relevant set* methods, and then call corrupt.
Some important points (mostly for the *cognoscenti*):

- One call to observe generates one scan (all rows have the same SCAN_NUMBER).
- The start and stop times specified to observe need not be contiguous and so one can simulate antenna drive times.
- Currently there is no facility for patterns of observing, such as mosaicing, since it is easy to do this via sequences of calls of observe.
- The heavy duty columns (DATA, FLAG, IMAGING_WEIGHT, *etc.* are tiled. New tiles are generated for each scan. Thus the TSM files will not get very large.

newsimulator changes some columns to the MeasurementSet to store results of processing. The following columns in the MS are particularly important:

DATA The original observed visibilities are in a column called DATA. These are normally not altered by any processing in AIPS++. However, this simulation program does overwrite these values.

CORRECTED_DATA During a calibration process, as carried out by *e.g.* calibrater, the visibilities may be corrected for calibration effects. This corrected visibilities are stored in a column CORRECTED_DATA which is created on demand.

MODEL_DATA During various phases of processing, the visibilities as predicted from some model are required. These model visibilities are stored in a column MODEL_DATA. The ft function of the imager tool should be used to calculate the model visibility for a model image or componentmodels.

Standard tools such as the table module and the ms can be used to access and possibly change these (and all other) columns.

Tools

newsimhelper	Tool to help newsimulator
newsimulator	Tool for simulation

1.8.1 newsimulator - Tool

Package synthesis
Module newsimulator

Tool for simulation

Description
newsimulator is a tool that performs simulation of synthesis data, including (optionally) creation of a MeasurementSet, prediction of model data, and corruption by various physical effects.

Constructors

newsimulator	Construct a newsimulator tool
newsimulatorfromms	Construct a newsimulator tool from a MS

Functions

Done	Close the newsimulator tool
close	Close the newsimulator tool
corrupt	Corrupt the data with visibility errors
name	Provide the name of the attached MeasurementSet
observe	Observe a given configuration
predict	Predict astronomical data from an image
reset	Reset the corruption terms
setauto	Set autocorrelation weight
setbandpass	Set the bandpasses
setconfig	Set the antenna configuration
setdata	Set the data parameters selection for subsequent processing
setfeed	Set the feed parameters
setfield	Set one or more observed fields
setgain	Set the gains
setknownconfig	Set the antenna configuration to a known array
setleakage	Set the polarization leakage
setlimits	Set limits for observing
setmosaicfield	Set observed mosaic fields
setnoise	Set the noise level
setoptions	Set various processing options
setpa	Corrupt phase by the parallactic angle
setseed	Set the seed for the random number generator
setspwindow	Set one or more spectral windows
settimes	Set integration time, *etc.*
setvp	Set the voltage pattern model for subsequent processing
summary	Summarize the current state
type	Return the type of this tool

newsimulator - Constructor

Package	synthesis
Module	newsimulator
Tool	newsimulator

Construct a newsimulator tool

Synopsis
newsimulator(ms)

Description
This is used to construct newsimulator tools. A newsimulator

tool can either be instantiated from an existing MeasurementSet, predicting and/or corrupting data on the given coordinates, or it can be used to create a fundamentally new MeasurementSet from descriptions of the array configuration and the observational parameters. This is useful for making a newsimulator tool which will make a MeasurementSet from scratch. In order to do this, you must also run setconfig, setfield, setspwindow, setfeed, and settimes. Creating the actual MS is performed by observe. Data can be predict-ed and then corrupted-ed. In this example, we read in the antenna coordinates from an ASCII file:

Arguments

ms	MeasurementSet to be created AllowedTable name Default:'MS'

Returns
newsimulator tool

Example

```
  tabname := 'VLAC.LOCAL.TAB';  asciifile := 'VLAC.LOCAL.STN'
  mytab := tablefromascii(tabname, asciifile);
  xx:=[]; yy:=[]; zz:=[]; diam:=[];
  xx := mytab.getcol('X');
  yy := mytab.getcol('Y');
  zz := mytab.getcol('Z');
  diam := mytab.getcol('DIAM');
#
  mysim := newsimulator('NEW1.ms');
# do configuration
  posvla := dm.observatory('vla');  #  dm.observatory('ALMA') also works!
  mysim.setconfig(telescopename='VLA', x=xx, y=yy, z=zz, dishdiameter=diam,
        mount='alt-az', antname='VLA',
                coordsystem='local', referencelocation=posvla);

# Initialize the spectral windows
  mysim.setspwindow(spwname='CBand', freq='5GHz',
    deltafreq='50MHz',
    freqresolution='50MHz',
    nchannels=1,
    stokes='RR RL LR LL');
```

```
  mysim.setspwindow(spwname='LBand', freq='1.420GHz',
    deltafreq='3.2MHz',
    freqresolution='3.2MHz',
    nchannels=32,
    stokes='RR LL');


# Initialize the source and calibrater
  mysim.setfield(sourcename='My cal',
 sourcedirection=dm.direction('J2000','00h0m0.0','+45.0.0.000'),
 calcode='A');
  mysim.setfield(sourcename='My source',
 sourcedirection=dm.direction('J2000','01h0m0.0','+47.0.0.000'));

  mysim.setlimits(shadowlimit=0.001, elevationlimit='8.0deg');
  mysim.setauto(autocorrwt=0.0);

  mysim.settimes(integrationtime='10s', usehourangle=F, referencetime=dm.epoch('utc',

  mysim.observe('My cal', 'LBand', starttime='0s', stoptime='300s');
  mysim.observe('My source', 'LBand', starttime='310s', stoptime='720s');
  mysim.observe('My cal', 'CBand', starttime='720s', stoptime='1020s');
  mysim.observe('My source', 'CBand', starttime='1030s', stoptime='1500s');

  mysim.setdata(spwid=1, fieldid=1);
  mysim.predict(modelimage='M31.MOD');
  mysim.setdata(spwid=2, fieldid=2);
  mysim.predict(modelimage='BigLBand.MOD');
  mysim.close();
```

---

newsimulatorfromms - Constructor

```
                          Package  synthesis
                           Module  newsimulator
                             Tool  newsimulator


Construct a newsimulator tool from a MS


Synopsis
newsimulatorfromms(ms)


Description
```

This is used to construct newsimulator tools operating on an existing MS. Data can be predicted and/or corrupted on the MS's given coordinates.

Arguments

ms	MeasurementSet to be processed AllowedTable name Default?'MS'

Returns
newsimulator tool

Example

```
mysim := newsimulatorfrommms('3C273XC1.MS');
mysim.predict('3C273XC1.modelImage');
mysim.setnoise(simplenoise='10mJy');
mysim.setgain(interval='100s', amplitude=[0.01, 0.01]);
mysim.corrupt();
mysm.close();
```

---

newsimulator.close - Function

Package synthesis
Module newsimulator
Tool newsimulator

Close the newsimulator tool

Synopsis
close()

Description
This is used to close newsimulator tools. Note that the data is written to disk. This is a synonym for done.

Returns
Bool

---

newsimulator.Done - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Close the newsimulator tool

Synopsis
Done()

Description
This is used to close and newsimulator tools. Note that the data is written to disk. This is a synonym for close.

Returns
Bool

---

newsimulator.name - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Provide the name of the attached MeasurementSet

Synopsis
name()

Description
Returns the name of the attached MeasurementSet.

Returns
String

---

newsimulator.summary - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Summarize the current state

Synopsis
summary()

Description
Writes a summary of the properties of the newsimulator to the default logger.

Returns
Bool

---

newsimulator.type - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Return the type of this tool

Synopsis
type()

Returns
String or fail

Description
This function returns the string 'newsimulator'. It is used so that in a script, you can make sure this *Glish* variable is a newsimulator tool.

---

newsimulator.settimes - Function

Package synthesis
Module newsimulator
Tool newsimulator

Set integration time, *etc.*

Synopsis
settimes(integrationtime, usehourangle, referencetime) geg

Description
The start and stop times are referenced to referencetime. Use either starttime/stoptime or startha/stopha. If the hour angles are specified, then the start and stop times are calculated such that the start time is later than the reference time, but less than one day later. The hour angles refer to the first source observed.

Arguments

integrationtime	Integration time AllowedQuantity Default:'10s'
usehourangle	Use starttime/stoptime as hour angles - else they are referenced to referencetime AllowedBool DefaultT
referencetime	Reference time for starttime and stoptime AllowedEpoch Default:'50000d'

---

newsimulator.observe - Function

Package synthesis
Module newsimulator
Tool newsimulator

Observe a given configuration

Synopsis
observe(sourcename, spwname, starttime, stoptime)

Description
Observe a given source with a given spectral window for the specified times, including start, stop, integration, and gap times. The start and stop times are referenced to referencetime. Use either starttime/stoptime or startha/stopha. If the hour angles are specified, then the start and stop times are calculated such that the start time is later than the reference time, but less than one day later. The hour angles refer to the first source observed.

Arguments

sourcename	Name of source or field (must be specified) AllowedString DefaultNone
spwname	Unique user-supplied name for this spectral window AllowedString DefaultNone
starttime	Start time referenced to referenceepoch AllowedQuantity Default:'0s'
stoptime	Stop time referenced to referenceepoch AllowedQuantity Default:'3600s'

---

newsimulator.setlimits - Function

Package synthesis
Module newsimulator
Tool newsimulator

Set limits for observing

Synopsis
setlimits(shadowlimit, elevationlimit)

Description

Data are flagged for two conditions:

Below elevation limit If either of the antennas point below the specified elevation limit then the data are flagged. The elevation is calculated correctly for antennas at different locations (such as occurs in VLBI).

Shadowing If one antenna shadows another such that the fractional (geometric) blockage is greater than the specified limit then the data are flagged. No correction for blockage is made for shadowed but non-flagged points.

Arguments

shadowlimit	Maximum fraction of geometrically shadowed area before flagging occurs AllowedDouble Default1e-6
elevationlimit	Minimum elevation angle before flagging occurs AllowedQuantity Default:'10deg'

---

newsimulator.setauto - Function

Package synthesis
Module newsimulator
Tool newsimulator

Set autocorrelation weight

Synopsis
setauto(autocorrwt)

Description

Arguments

autocorrwt	Weight to assign autocorrelations (0=none) AllowedFloat Default0.0

---

newsimulator.setconfig - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set the antenna configuration

Synopsis
setconfig(telescopename, x, y, z, dishdiameter, mount, antname, coordsystem, referencelocation)

Description
Set the positions of the antennas. Note that the name of the telescope will control which voltage pattern is applied to the data.

Arguments

telescopename	Name of the telescope we are simulating (determines VP) AllowedString Default:'VLA'
x	Vector of x values of all antennas [currently m] AllowedVector Default:[]
y	Vector of y values of all antennas [currently m] AllowedVector Default:[]
z	Vector of z values of all antennas [currently m] AllowedVector Default:[]
dishdiameter	Vector of diameters of all antennas [currently m] AllowedVector Default:[]
mount	Vector of mount types of all antennas (recognized mounts are 'ALT-AZ', 'EQUATORIAL', 'X-Y', 'ORBITING', 'BIZARRE' AllowedVector of strings Default:[]
antname	Vector of names of all antennas AllowedVector of strings Default:[]
coordsystem	Coordinate system of antenna positions [x,y,z] AllowedString: 'global' | 'local' | 'longlat' Default:'global'
referencelocation	Reference location [required for local coords] AllowedMPosition

Example

```
diam := [25, 25, 25, 25, 25]
xx := [50, 100, 150, 200, 250]
yy := [2, -5, -20, -50, -100]
zz := [-0.5, -1.0, -1.5, -2.0, -2.5]
posvla := dm.observatory('vla');
mysim.setconfig(telescopename='VLA', x=xx, y=yy, z=zz, dishdiameter=diam,
               mount='alt-az', antname='VLA',
```

```
                coordsystem='local', referencelocation=posvla);
```

In this example we configure an array of 5 dishes set at the VLA position. We use the measures tool to give us the location of the vla.

---

newsimulator.setknownconfig - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set the antenna configuration to a known array

Synopsis
setknownconfig(arrayname)

Description
Sets the configuration to a known array such as VLAA, VLBA, EVN or ATCA6.0A. The arrays are those known to newsimhelper. All the information needed by setconfig is filled in.

Arguments

arrayname	Name of the telescope configurationwe are simulating AllowedString Default?'VLAA'

Example

```
 mysim.setknownconfig('ATCA6.0A');
```

In this example we configure an array of 5 dishes set at the VLA position. We use the measures tool to give us the location of the vla.

---

newsimulator.setfeed - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set the feed parameters

Synopsis
setfeed(mode)

Description
The goal is to let the feed parameters be specified for each antenna and each spectral window. At this moment, you only have the choice between 'perfect R L' and 'perfect X Y' (i.e., you cannot invent your own corrupted feeds yet). Doesn't need to be run if you want perfect R and L feeds.

Arguments

mode	Mode for specifying feed parameters (currently, perfect only) AllowedString Default:'perfect R L'

---

newsimulator.setfield - Function

Package synthesis
Module newsimulator
Tool newsimulator

Set one or more observed fields

Synopsis
setfield(sourcename, sourcedirection, calcode, distance)

Description
Set one or more observed fields, including name, coordinates, calibration code. Can be invoked multiple times for a complex observation. Must be invoked at least once before observe.
If the distance to the object is set then the phase term includes a curvature for the near-field effect at the center of the image.

Arguments

sourcename	Name of source or field (must be specified) AllowedString Default:'unknown'
sourcedirection	Coordinates of source to be observed AllowedMDirection
calcode	Calibration code AllowedString Default:''
distance	Distance to the object AllowedString Default:'0km'

Example

```
mysim.setconfig(telescopename='VLA', x=xx, y=yy, z=zz, dishdiameter=diam,
              mount='alt-az', antname='VLA',
              coordsystem='local', referencelocation=dm.observatory('vla'));

mysim.setspwindow(spwname='XBAND', freq='8GHz', deltafreq='50MHz',
                  freqresolution='50MHz', nchannels=1, stokes='RR
                  LL');
dir0 := dm.direction('B1950',  '16h00m0.0', '50d0m0.000')
mysim.setfield(sourcename='SIMU1', sourcedirection=dir0,
              integrations=1, xmospointings=1, ymospointings=1,
              mosspacing=1.0);
mysim.observe('SIMU1', 'XBAND', integrationtime='10s', usehourangle=T,
              starttime='0s', stoptime='3600s',
              referencetime=reftime);
```

---

newsimulator.setmosaicfield - Function

```
Package  synthesis
 Module  newsimulator
   Tool  newsimulator
```

Set observed mosaic fields

Synopsis
setmosaicfield(sourcename, calcode, fieldcenter, xmosp, ymosp,
mosspacing, distance)

Description

Set mosaic fields by internally invoking setfield multiple times. Currently only handle a rectangular mosaicing pattern. Either setfield or setmosaicfield must be invoked at least once before observe.

If the distance to the object is set then the phase term includes a curvature for the near-field effect at the center of the image.

Arguments

sourcename	Name of source or field (must be specified). AllowedString Default:'unknown'
calcode	Calibration code AllowedString Default:''
fieldcenter	Coordinates of mosaic field center AllowedMDirection
xmosp	Number of mosaic pointing in horizontal direction AllowedInt Default1
ymosp	Number of mosaic pointing in vertical direction AllowedInt Default1
mosspacing	Spacing between mosaic pointings AllowedQuantity Default:'1arcsec'
distance	Distance to the object AllowedString Default:'0km'

Example

```
mysim.setconfig(telescopename='VLA', x=xx, y=yy, z=zz, dishdiameter=diam,
              mount='alt-az', antname='VLA',
              coordsystem='local', referencelocation=dm.observatory('vla'));

mysim.setspwindow(spwname='XBAND', freq='8GHz', deltafreq='50MHz',
                freqresolution='50MHz', nchannels=1, stokes='RR
                LL');
dir0 := dm.direction('B1950',  '16h00m0.0', '50d0m0.000')
```

```
  mysim.setmosaicfield(sourcename='SIMU1', fieldcenter=dir0,
                       xmosp=2, ymosp=2, mosspacing='154.5arcsec');
  mysim.settimes(integrationtime='10s');
  mysim.observe('SIMU1_1', 'XBAND', starttime='0s', stoptime='100s');
  mysim.observe('SIMU1_2', 'XBAND', starttime='110s', stoptime='210s');
  mysim.observe('SIMU1_3', 'XBAND', starttime='220s', stoptime='320s');
  mysim.observe('SIMU1_4', 'XBAND', starttime='330s', stoptime='430s');
```

In this example, a 2x2 mosaic pattern is set centered around the direction specified (dir0). The setmosaicfield name each mosaic field with i sourcename+'_i', where i= 1..4.

---

newsimulator.setspwindow - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set one or more spectral windows

Synopsis

setspwindow(spwname, freq, deltafreq, freqresolution, nchannels, stokes)

Description

Set one or more spectral windows for the observations, including starting frequency, number of channels, channel increment and resolution, and stokes parameters observed. Can be invoked multiple times for a complex observation. Must be invoked at least once before observe.

Arguments

spwname	Unique user-supplied name for this spectral window AllowedString Default:'XBAND'
freq	Starting frequency AllowedQuantity Default:'8.0GHz'
deltafreq	Frequency increment per channel AllowedQuantity Default:'50.0MHz'
freqresolution	Frequency resolution per channel AllowedQuantity Default:'50.0MHz'
nchannels	Number of channels AllowedInt Default1
stokes	Stokes types to simulate AllowedString Default:'RR LL'

Example

To simulate a two spectral window (or two IF's in VLA jargon) data set, use setpwid as follows (here we are simulating 16 channels, 50MHz wide channel for each spectral window)

```
mysim.setspwindow(spwname='CBAND', freq='2GHz', deltafreq='50MHz',
                  freqresolution='50MHz', nchannels=16, stokes='RR LL');

mysim.setspwindow(spwname='SBAND', freq='5GHz', deltafreq='50MHz',
                  freqresolution='50MHz', nchannels=16, stokes='RR LL');
```

Note that the spwname is used in observe to determine which spectral window is used.

---

newsimulator.setdata - Function

```
Package  synthesis
 Module  newsimulator
   Tool  newsimulator
```

Set the data parameters selection for subsequent processing

Synopsis
setdata(spwid, fieldid, msselect, async)

Description
This setup tool function selects which data are to be used subsequently. After invocation of setdata, only the selected data are operated on.

Arguments

spwid	Spectral Window Ids (1 relative) to select AllowedVector of Ints Default:1
fieldid	Field Ids (1 relative) to select AllowedVector of Ints Default:1
msselect	TQL select string applied as a logical "and" with the other selections AllowedString
async	Run asynchronously in the background? AllowedBool Default:dowait

---

newsimulator.predict - Function

Package synthesis
Module newsimulator
Tool newsimulator

Predict astronomical data from an image

Synopsis
predict(modelImage, complist, incremental)

Description
Predict astronomical data from an image. The (u,v) coordinates already exist, either from a MeasurementSet we have read in or by generating the MeasurementSet coordinates and empty data through create(). We simply predict onto these coordinates.

Arguments

modelImage	Name of image from which to predict visibilities AllowedString Default:''
complist	Name of component list AllowedString
incremental	Add this model to the existing Data Visibilities? AllowedBool DefaultF

---

newsimulator.setoptions - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set various processing options

Synopsis
setoptions(ftmachine, cache, tile, gridfunction, location, padding, facets, maxdata)

Description

For most of these, set the options for predict details. See also imager help for more details.

To simulate single dish data, use gridft=SD and gridfunction=PB.

Arguments

ftmachine	Fourier transform machine AllowedString: 'gridft'|'SD' Default:'gridft'
cache	Size of gridding cache in complex pixels AllowedInt Default0
tile	Size of a gridding tile in pixels (in 1 dimension) AllowedInt Default16
gridfunction	Gridding function AllowedString: 'SF'|'BOX'|'PB' Default:'SF'
location	Location used in phase rotations Allowedposition measure
padding	Padding factor in image plane (>=1.0) AllowedFloat Default1.3
facets	Number of facets AllowedInt Default1
maxdata	Maximum data to write to a single TSM file (MB) AllowedDouble Default2000

Returns
Bool

Example

```
mysim.setoptions(cache=10000000, tile=32,
gridfunction='BOX', dm.location('vla'))
```

---

newsimulator.setvp - Function

Package synthesis
Module newsimulator
Tool newsimulator

Set the voltage pattern model for subsequent processing

Synopsis
setvp(dovp, usedefaultvp, vptable, dosquint, parangleinc)

Description
Set the voltage pattern model (and hence, the primary beam) used for a Telecope. There are currently two ways to set the voltage pattern: by using the extensive list of defaults which the system knows about, or by creating a voltage pattern description with the vpmanager. The default voltage patterns include both a high and a low frequency VP for the WSRT, a VP for each observing band at the AT, several VP's for the VLA, including the approrpiate beam squint for each observing band, and Gaussian for the BIMA dishes. If you are simulating a telescope which doesn't yet exist, you will need to supply a model voltage pattern using the vpmanager.

Arguments

dovp	Multiply by the voltage pattern (ie, primary beam) when simulating AllowedBool DefaultT
usedefaultvp	Look up the default VP for this telescope and frequency? AllowedBool DefaultT
vptable	If usedefaultvp is false, provide a VP Table made with vpmanager AllowedTable
dosquint	Activate the beam squint in the VP model AllowedBool DefaultT
parangleinc	Parallactice angle increment for squint application AllowedQuantity Default360deg

Returns
Bool

Example

```
imgr.setvp(dovp=T, usedefaultvp=F,
vptable='MyAlternateVLAPBModel.TAB', dosquint=F);
```

---

newsimulator.corrupt - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Corrupt the data with visibility errors

Synopsis
corrupt()

Description
Add errors specified by the set functions (such as noise, gains, polarization leakage, bandpass, etc) to the synthesis data. This is only visibility-based effects, not image-plane effects such as pointing errors and voltage patterns, which get applied in predict.

Example

```
MSNewsimulator mysim('3C273XC1.MS');
mysim.predict('3C273XC1.FAKE.IMAGE');
mysim.setnoise( mode='simplenoise', simplenoise='0.1Jy');
mysim.setpa( mode='calculate');
mysim.corrupt();
```

---

newsimulator.reset - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Reset the corruption terms

Synopsis
reset()

Description
Reset the visibility corruption terms: this means that corrupt introduces no errors.

---

newsimulator.setbandpass - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set the bandpasses

Synopsis
setbandpass(mode, table, interval, amplitude)

Description
Set the level of bandpass errors. The error distributions are normal, mean zero, with the variances as specified. (Not yet implemented).

Arguments

mode	Mode of operation AllowedString: 'calculate'|'table' Default:'calculate'
table	Name of table AllowedString Default:''
interval	Coherence interval e.g. '1h' AllowedQuantity Default:'1h'
amplitude	Variances errors in amplitude and phase Allowed2-Vector of Double Default:[0, 0]

---

newsimulator.setgain - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set the gains

Synopsis
setgain(mode, table, interval, amplitude)

Description
Set the level of gain errors. The error distributions are normal, mean zero, with the variances as specified.

Arguments

mode	Mode of operation AllowedString: 'calculate'|'table' Default:'calculate'
table	Name of table AllowedString Default:''
interval	Coherence interval e.g. '10s' AllowedQuantity Default:'10s'
amplitude	Variances of errors in amplitude and phase Allowed2-Vector of Double Default[0, 0]

---

newsimulator.setleakage - Function

Package	synthesis
Module	newsimulator
Tool	newsimulator

Set the polarization leakage

Synopsis
setleakage(mode, table, interval, amplitude)

Description
Set the level of polarization leakage between feeds.

Arguments

mode	Mode of operation AllowedString: 'calculate'|'table' Default:'calculate'
table	Name of table AllowedString Default:''
interval	Coherence interval e.g. '5.0h' AllowedQuantity Default:'5.0h'
amplitude	Variance of pol leakage amplitude errors; currently, just real AllowedDouble Default0.0

---

newsimulator.setnoise - Function

Package synthesis
Module newsimulator
Tool newsimulator

Set the noise level

Synopsis
setnoise(mode, table, simplenoise, antefficiency, correfficiency, spillefficiency, tau, trx, tatmos, tcmb)

Description
Set various system parameters from which the thermal (ie, random additive) noise level will be calculated. The noise will vary with dish diameter, antenna efficiency, system temperature, opacity, sky temperature, etc. The noise will increase with the airmass if tau is greater than zero. The noise is calculated according to the *Brown Equation* (ie, R.L. Brown's calculation of MMA sensitivity, 30ct95):

$$\Delta S = \frac{4\sqrt{2}[T_{rx}e^{\tau A} + T_{atm}(e^{\tau A} - \epsilon_l) + T_{cmb}]}{\epsilon_q \epsilon_a \pi D^2 \sqrt{\Delta\nu \Delta t}} \tag{1.37}$$

Arguments

mode	Mode of operation AllowedString: 'table'|'simplenoise'|'calculate' Default:'calculate'
table	Name of noise table if not simplenoise or not calculated AllowedString Default:''
simplenoise	Level of noise if not Table or not Calculated AllowedQuantity Default:'0.0Jy'
antefficiency	antenna efficiency AllowedFloat Default0.8
correfficiency	Correlation efficiency AllowedFloat Default0.85
spillefficiency	Forward spillover efficiency AllowedFloat Default0.85
tau	Atmospheric Opacity AllowedFloat Default0.0
trx	Receiver temp (ie, all non-atmospheric Tsys contributions) [K] AllowedFloat Default50
tatmos	Temperature of atmosphere [k] AllowedFloat Default250.0
tcmb	Temperature of cosmic microwave background [K] AllowedFloat Default2.7

---

newsimulator.setpa - Function

Package synthesis
Module newsimulator
Tool newsimulator

Corrupt phase by the parallactic angle

Synopsis
setpa(mode, table, interval)

Description
Corrupt phase by the parallactic angle

Arguments

mode	Mode of operation AllowedString: 'calculate'|'table' Default:'calculate'
table	Name of table AllowedString Default:''
interval	Interval for parallactic angle application, e.g. '10s' AllowedQuantity Default:'10s'

---

newsimulator.setseed - Function

Package synthesis
Module newsimulator
Tool newsimulator

Set the seed for the random number generator

Synopsis
setseed(seed)

Description

Arguments

seed	Seed AllowedInt Default185349251

1.8.2 newsimhelper - Tool

Package synthesis
Module newsimulator

Tool to help newsimulator

Description
newsimhelper helps newsimulator by providing information such as antenna locations in a convenient form. newsimhelper usually need not be called directly.

Constructors

newsimhelper Construct a newsimhelper tool

Functions

close	Close the newsimhelper tool
getallstations	Get all stations known to newsimhelper
getarray	Get a specific array
getarrays	Get all arrays known to newsimhelper
type	Return the type of this tool

newsimhelper - Constructor

Package synthesis
Module newsimulator
Tool newsimhelper

Construct a newsimhelper tool

Synopsis
newsimhelper(schedtable)

Description
This is used to construct newsimhelper tools. An optional argument is the name of a table containing the antenna locations drawn from Craig Walker's SCHED data file. Usually the table in the AIPS++ data repository (geodetic/SCHED_locations) is adequate and should be used (the default).

Arguments

schedtable	Table of antenna locations AllowedTable Defaultunset

Returns
newsimhelper tool

Example

```
include 'newsimhelper.g';
- ns:=newsimhelper()
- ns.getarrays()
VLAA VLAB VLAC VLAD VLBA EVN ATCA6.0A ATCA6.0B ATCA6.0C ATCA6.0D
ATCA1.5A ATCA1.5B ATCA1.5C ATCA1.5D ATCA0.75A ATCA0.75B ATCA0.75C
ATCA0.75D ATCA0.375 ATCA0.210 ATCA0.122A ATCA0.122B ATCA0.244
- ns.getarray('ATCA0.244')
[names=ATCA_01 ATCA_03 ATCA_05 ATCA_07 ATCA_09 ATCA_37, diam=[22 22 22
22 22 22] , telescope=ATCA0point244, x=[-4.75245e+06 -4.75242e+06
-4.75239e+06 -4.75235e+06 -4.75232e+06 -4.7494e+06] , y=[2.79033e+06
2.79038e+06 2.79043e+06 2.79048e+06 2.79054e+06 2.7955e+06] ,
z=[-3.20049e+06 -3.20049e+06 -3.20049e+06 -3.20049e+06 -3.20049e+06
-3.20049e+06] , mount=alt-az alt-az alt-az alt-az alt-az alt-az,
offset=[0 0 0 0 0 0] ]
```

---

newsimhelper.close - Function

Package	synthesis
Module	newsimulator
Tool	newsimhelper

Close the newsimhelper tool

Synopsis
close()

Description
This is used to close newsimhelper tools.

Returns
Bool

---

newsimhelper.type - Function

Package	synthesis
Module	newsimulator
Tool	newsimhelper

Return the type of this tool

Synopsis
type()

Returns
String or fail

Description
This function returns the string 'newsimhelper'. It is used so that in a script, you can make sure this *Glish* variable is a newsimhelper tool.

---

newsimhelper.getarrays - Function

Package	synthesis
Module	newsimulator
Tool	newsimhelper

Get all arrays known to newsimhelper

Synopsis
getarrays()

Description
newsimhelper contains definitions of various arrays. These are defined in terms of the names (the DBNAME column) in the SCHED_locations file located in the data repository geodetic area. The names currently known are:

VLAA VLAB VLAC VLAD VLBA EVN ATCA6.0A ATCA6.0B ATCA6.0C ATCA6.0D
ATCA1.5A ATCA1.5B ATCA1.5C ATCA1.5D ATCA0.75A ATCA0.75B ATCA0.75C
ATCA0.75D ATCA0.375 ATCA0.210 ATCA0.122A ATCA0.122B ATCA0.244

---

newsimhelper.getarray - Function

Package	synthesis
Module	newsimulator
Tool	newsimhelper

Get a specific array

Synopsis
getarray(arrayname)

Description
newsimhelper contains definitions of various arrays. These are
defined in terms of the names (the DBNAME column) in the
SCHED_locations file located in the data repository geodetic
area. The names currently known are:
VLAA VLAB VLAC VLAD VLBA EVN ATCA6.0A ATCA6.0B ATCA6.0C ATCA6.0D
ATCA1.5A ATCA1.5B ATCA1.5C ATCA1.5D ATCA0.75A ATCA0.75B ATCA0.75C
ATCA0.75D ATCA0.375 ATCA0.210 ATCA0.122A ATCA0.122B ATCA0.244
getarray returns a record containing the information in a format
ready to be used in newsimulator.

Arguments

arrayname	Name of array
	AllowedString
	Default?'VLAD'

Returns
newsimhelper tool

Example

```
include 'newsimhelper.g';
- ns:=newsimhelper()
```

```
- ns.getarrays()
VLAA VLAB VLAC VLAD VLBA EVN ATCA6.0A ATCA6.0B ATCA6.0C ATCA6.0D
ATCA1.5A ATCA1.5B ATCA1.5C ATCA1.5D ATCA0.75A ATCA0.75B ATCA0.75C
ATCA0.75D ATCA0.375 ATCA0.210 ATCA0.122A ATCA0.122B ATCA0.244
- ns.getarray('ATCA0.244')
[names=ATCA_01 ATCA_03 ATCA_05 ATCA_07 ATCA_09 ATCA_37, diam=[22 22 22
22 22 22] , telescope=ATCA0point244, x=[-4.75245e+06 -4.75242e+06
-4.75239e+06 -4.75235e+06 -4.75232e+06 -4.7494e+06] , y=[2.79033e+06
2.79038e+06 2.79043e+06 2.79048e+06 2.79054e+06 2.7955e+06] ,
z=[-3.20049e+06 -3.20049e+06 -3.20049e+06 -3.20049e+06 -3.20049e+06
-3.20049e+06] , mount=alt-az alt-az alt-az alt-az alt-az alt-az,
offset=[0 0 0 0 0 0] ]
```

---

newsimhelper.getallstations - Function

```
                                        Package  synthesis
                                         Module  newsimulator
                                           Tool  newsimhelper


Get all stations known to newsimhelper


Synopsis
getallstations()


Description
newsimhelper contains definitions of various arrays.  These are
defined in terms of the names (the DBNAME column) in the
SCHED_locations file located in the data repository geodetic
area.  The names currently known are:
VLAA VLAB VLAC VLAD VLBA EVN ATCA6.0A ATCA6.0B ATCA6.0C ATCA6.0D
ATCA1.5A ATCA1.5B ATCA1.5C ATCA1.5D ATCA0.75A ATCA0.75B ATCA0.75C
ATCA0.75D ATCA0.375 ATCA0.210 ATCA0.122A ATCA0.122B ATCA0.244
getallstations returns a string containing all station names
known to newsimhelper.


Arguments



Returns
String
```

Example

```
include 'newsimhelper.g';
- ns:=newsimhelper()
- ns.getallstations()
ALGOPARK ARECIBO ATCA_01 ATCA_02 ATCA_03 ATCA_04 ATCA_05 ATCA_06
ATCA_07 ATCA_08 ATCA_09 ATCA_10 ATCA_104 ATCA_104 ATCA_11 ATCA_12
ATCA_13 ATCA_14 ATCA_15 ATCA_16 ATCA_17 ATCA_18 ATCA_19 ATCA_20
ATCA_21 ATCA_22 ATCA_23 ATCA_24 ATCA_25 ATCA_26 ATCA_27 ATCA_28
ATCA_29 ATCA_30 ATCA_31 ATCA_32 ATCA_33 ATCA_34 ATCA_35 ATCA_36
ATCA_37 AZORES BLKBUTTE BR-VLBA BREST CAMB32 CARNUSTY CHLBOLTN CRIMEA
DS14 DSS14 DSS15 DSS43 DSS45 DSS63 DSS65 EFLSBERG ELY FD-VLBA FORTLEZA
FORTORD FORTORDS GBANK_TS GBT GGAO7108 GILCREEK GO GOLDMARS GOLDS_TS
GOLDVENU GORF7102 GRASSE HALEAKAL HARTRAO HATCREEK HAYSTACK HN-VLBA
HOBART26 HOFN HOHENFRG HOHNBERG HRAS_085 JB_Mk_2 JODLMKI KALYAZIN
KARLBURG KASHIM34 KASHIMA KAUAI KOKEE KP-VLBA KP12 KWAJAL26 LA-VLBA
MADRD_TS MADRID64 MARCUS MARPOINT MATERA MEDICINA METSAHOV MIZNAO10
MK-VLBA MOJAVE12 MOPRA NL-VLBA NOBEY_6M NOTO NRAO140 NRAO20 NRAO85_1
NRAO85_3 NYALES20 OHIGGINS ONSALA60 ONSALA85 OV-VLBA OVRO_130 OVRO_MMA
PARKES PENTICTN PICOVEL PIETOWN PLATTVIL PRESIDIO PT_REYES RICHMOND
ROBLD_TS ROBLED32 SANTIA12 SC-VLBA SESHAN25 SEST SHANGHAI TDBIN_TS
TIDBIN64 TIGOCONC TIGOWTZL TORUN TOULOUSE TROMSONO TRYSILNO TSUKUBA
URUMQI USSURIYSK USUDA_64 USUDA_TS VLA VLA_E1 VLA_E10 VLA_E12 VLA_E14
VLA_E16 VLA_E18 VLA_E2 VLA_E20 VLA_E24 VLA_E28 VLA_E3 VLA_E32 VLA_E36
VLA_E4 VLA_E40 VLA_E48 VLA_E5 VLA_E56 VLA_E6 VLA_E64 VLA_E7 VLA_E72
VLA_E8 VLA_E9 VLA_MAS VLA_N1 VLA_N10 VLA_N12 VLA_N14 VLA_N16 VLA_N18
VLA_N2 VLA_N20 VLA_N24 VLA_N28 VLA_N3 VLA_N32 VLA_N36 VLA_N4 VLA_N40
VLA_N48 VLA_N5 VLA_N56 VLA_N6 VLA_N64 VLA_N7 VLA_N72 VLA_N8 VLA_N9
VLA_OUT VLA_W1 VLA_W10 VLA_W12 VLA_W14 VLA_W16 VLA_W18 VLA_W2 VLA_W20
VLA_W24 VLA_W28 VLA_W3 VLA_W32 VLA_W36 VLA_W4 VLA_W40 VLA_W48 VLA_W5
VLA_W56 VLA_W6 VLA_W64 VLA_W7 VLA_W72 VLA_W8 VLA_W9 VNDNBERG WB_RT0
WB_RT1 WB_RT2 WB_RT3 WB_RT4 WB_RT5 WB_RT6 WB_RT7 WB_RT8 WB_RT9 WB_Tied
WESTFORD WETTZELL YEBES YELLOWKN YLOW7296
```