Atacama
Large
Millimeter /
submillimeter
Array

ALMA Science Operations:
Cycle 6 Pipeline Reference Manual

Version: A.1

2018-10-01

Prepared by:	Organization:	Signature
D. Muders, K. Sugimoto	ALMA	
Approved by:		
R. Indebetouw	ALMA	

List of Heuristics Tasks (Pipeline-CASA54-P1-B-r42028)

Generic

13 tasks available.

task name	description
h_applycal	Apply the calibration(s) to the data
h_export_calstate	Save the pipeline calibration state to disk
h_exportdata	Prepare interferometry data for export
h_import_calstate	Import a calibration state from disk
h_importdata	Imports data into the interferometry pipeline
h_init	Initialize the interferometry pipeline
h_mssplit	Select data from calibrated MS(s) to form new MS(s) for imaging
h_restoredata	Restore flagged and calibration interferometry data from a pipeline run
h_resume	Restore a save pipeline state from disk
h_save	Save the pipeline state to disk
h_show_calstate	Show the current pipeline calibration state
h_tsyscal	Derive a Tsys calibration table
h_weblog	Open the pipeline weblog in a browser

Interferometry Generic

26 tasks available.

task name	description
hif_antpos	Derive an antenna position calibration table
hif_applycal	Apply the calibration(s) to the data
hif_atmflag	Flag channels with bad atmospheric transmission
hif_bandpass	Compute bandpass calibration solutions
hif_checkproductsize	Check imaging product size
hif_correctedampflag	Flag corrected - model amplitudes based on calibrators.
hif_editimlist	Add to a list of images to be produced with hif_makeimages()
hif_findcont	Find continuum frequency ranges
hif_gaincal	Determine temporal gains from calibrator observations
hif_gainflag	Flag antennas with deviant gain
hif_linfeedpolcal	Base linfeedpolcal task
hif_lowgainflag	Flag antennas with low or high gain
hif_makecutoutimages	Base makecutoutimages task
hif_makeimages	Compute clean map
hif_makeimlist	Compute list of clean images to be produced
hif_makermsimages	Base makermsimages task
hif_mstransform	Select data from calibrated MS(s) to form new MS(s) for imaging
hif_polarization	Base polarization task
hif_rawflagchans	Flag deviant baseline/channels in raw data
hif_refant	Select the best reference antennas
hif_setjy	Fill the model column with calibrated visibilities
hif_setmodels	Set calibrator source models
hif_tclean	Compute clean map
hif_transformimagedata	Base transformimagedata task
hif_uvcontfit	Fit the continuum in the UV plane
hif_uvcontsub	Subtract the fitted continuum from the data

Interferometry ALMA

22 tasks available.

task name	description
hifa_antpos	Derive an antenna position calibration table
hifa_bandpassflag	Compute bandpass calibration with flagging
hifa_bandpass	Compute bandpass calibration solutions
hifa_bpsolint	Compute optimal bandpass calibration solution intervals
hifa_exportdata	Prepare interferometry data for export
hifa_flagdata	Do basic flagging
hifa_flagtargets	Do science target flagging
hifa_fluxcalflag	Locate and flag line regions in solar system flux calibrators
hifa_fluxdb	Connect to flux calibrator database
hifa_gaincalsnr	Compute gaincal signal to noise ratios per spw
hifa_gfluxscaleflag	Derive the flux density scale with flagging
hifa_gfluxscale	Derive flux density scales from standard calibrators
hifa_imageprecheck	Base imageprecheck task
hifa_importdata	Imports data into the interferometry pipeline
hifa_linpolcal	Compute polarization calibration
hifa_restoredata	Restore flagged and calibration interferometry data from a pipeline run
hifa_session_bandpass	Compute bandpass calibration solutions
hifa_spwphaseup	Compute phase calibration spw map and per spw phase offsets
hifa_timegaincal	Determine temporal gains from calibrator observations
hifa_tsysflag	Flag deviant system temperature measurements
hifa_wvrgcalflag	No description
hifa_wvrgcal	No description

Interferometry VLA

29 tasks available.

task name	description
hifv_applycals	Applycals
hifv_checkflag	Run flagdata in rflag mode
hifv_circfeedpolcal	Base circfeedpolcal task
hifv_exportdata	Prepare interferometry data for export
hifv_exportvlassdata	Base exportvlassdata task
hifv_finalcals	Finalcals
hifv_flagbaddef	Flagging of deformatters - amp and phase
hifv_flagcal	Base flagging task
hifv_flagdata	Do basic deterministic flagging of a list of MeasurementSets
hifv_fluxboot2	Fluxboot2
hifv_fluxboot	Fluxboot
hifv_gaincurves	Runs gencal in gc mode
hifv_hanning	Hanning smoothing on a dataset
hifv_importdata	Imports data into the VLA pipeline
hifv_opcal	Runs gencal in opac mode
hifv_pbcor	Base pbcor task
hifv_plotsummary	End of VLA pipeline plotsummary
hifv_priorcals	Runs gaincurves, opacities, requantizer gains, antenna position corrections, and tec_maps
hifv_restoredata	Restore flagged and calibration interferometry data from a pipeline run
hifv_rqcal	Runs gencal in rq mode
hifv_semiFinalBPdcals	Runs a second quick calibration to set up for heuristic flagging
hifv_solint	Determines different solution intervals
hifv_statwt	Statwt
hifv_swpowcal	Runs gencal in swpow mode

hifv_syspower	Base syspower task
hifv_targetflag	Targetflag
hifv_tecmaps	Base tecmaps task
hifv_testBPdcals	Runs initial delay calibration to set up heuristic flagging
hifv_vlasetjy	Does an initial setjy run on the vis

Single-Dish

11 tasks available.

task name	description
hsd_applycal	Apply the calibration(s) to the data
hsd_baseline	Detect and validate spectral lines, subtract baseline by masking detected lines
hsd_blflag	Flag spectra based on predefined criteria of single dish pipeline
hsd_exportdata	Prepare single dish data for export
hsd_flagdata	Do basic flagging of a list of MeasurementSets
hsd_imaging	Generate single dish images
hsd_importdata	Imports data into the single dish pipeline
hsd_k2jycal	Derive Kelvin to Jy calibration tables
hsd_restoredata	Restore flagged and calibration single dish data from a pipeline run
hsd_skycal	Calibrate data
hsd_tsysflag	Flag deviant system temperature measurements

Created at Fri Sep 14 10:04:26 2018 UTC
This page is automatically generated from XML interface definition files.

Summary of generic tasks and parameters

h_applycal

Task Description

Apply the calibration(s) to the data
Apply precomputed calibrations to the data.

---- pipeline parameter arguments which can be set in any pipeline mode
applymode -- Calibration apply mode
    ''='calflagstrict': calibrate data and apply flags from solutions using
        the strict flagging convention
    'trial': report on flags from solutions, dataset entirely unchanged
    'flagonly': apply flags from solutions only, data not calibrated
    'calonly': calibrate data only, flags from solutions NOT applied
    'calflagstrict':
    'flagonlystrict': same as above except flag spws for which calibration is
        unavailable in one or more tables (instead of allowing them to pass
        uncalibrated and unflagged)
  default: ''
flagsum -- Compute before and after flagging statistics summaries.
  default: True
flagdetailedsum -- Compute detailed before and after flagging statistics summaries
  if flagsum is True.
  default: False
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
  determines the values of all context defined pipeline inputs automatically.
  In interactive mode the user can set the pipeline context defined parameters
  manually. In 'getinputs' mode the user can check the settings of all
  pipeline parameters without running the task.
  default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
    in the pipeline context.
    default: []
    example: ['X227.ms']
field -- A string containing the list of field names or field ids to which
    the calibration will be applied. Defaults to all fields in the pipeline
    context.
    default: ''
    example: '3C279', '3C279, M82'
intent -- A string containing the list of intents against which the
    selected fields will be matched. Defaults to all supported intents
    in the pipeline context.
    default: ''
    example: '*TARGET*'
spw -- The list of spectral windows and channels to which the calibration
    will be applied. Defaults to all science windows in the pipeline
    context.
    default: ''
    example: '17', '11, 15'
antenna -- The list of antennas to which the calibration will be applied.
    Defaults to all antennas. Not currently supported.
--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned
Description
hif_applycal applies the precomputed calibration tables stored in the pipeline
context to the set of visibility files using predetermined field and
spectral window maps and default values for the interpolation schemes.
Users can interact with the pipeline calibration state using the tasks
hif_export_calstate and hif_import_calstate.
Issues
There is some discussion about the appropriate values of calwt. Given
properly scaled data, the correct value should be the CASA default of True.
However at the current time ALMA is suggesting that calwt be set to True for
applying observatory calibrations, e.g. antenna positions, WVR, and system
temperature corrections, and to False for applying instrument calibrations,
e.g. bandpass, gain, and flux.
Examples
1. Apply the calibration to the target data
hif_applycal (intent='TARGET')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
applymode	string	None	Calibration mode: ""="calflagstrict","calflag","calflagstrict","trial","flagonly","flagonlystrict", or "calonly"
flagbackup	bool	True	Backup the flags before the apply
flagsum	bool	True	Compute before and after flagging summary statistics
flagdetailedsum	bool	False	Compute before and after flagging summary statistics
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

h_export_calstate

Task Description

Save the pipeline calibration state to disk
h_export_calstate saves the current pipeline calibration state to disk
in the form of a set of equivalent applycal calls.
Keyword arguments:
filename -- Name for the saved calibration state.
state -- calibration state to export
Description
h_export_calstate saves the current pipeline calibration state to disk
in the form of a set of equivalent applycal calls.
If filename is not given, h_export_calstate saves the calibration state to
disk with a filename based on the pipeline context creation time, using the
extension '.calstate'
One of two calibration states can be exported: either the active calibration
state (those calibrations currently applied on-the-fly but scheduled for
permanent application to the MeasurementSet in a subsequent hif_applycal
call) or the applied calibration state (calibrations that were previously
applied to the MeasurementSet using hif_applycal). The default is to export
the active calibration state.
Issues

If run several times in one pipeline session does the automatic export
file naming scheme, overwrite previous versions?
Example
1. Save the calibration state.
   h_export_calstate()
2. Save the active calibration state with a custom filename
   h_export_calstate(filename='afterbandpass.calstate')
3. Save the applied calibration state with a custom filename
   h_export_calstate(filename='applied.calstate', state='applied')

Parameter List

name	type	default	description
filename	string	None	Name for saved calibration state
state	string	active	The calibration state to export

Back

---

h_exportdata

Task Description

Prepare interferometry data for export
The hif_exportdata task exports the data defined in the pipeline context
and exports it to the data products directory, converting and or
packing it as necessary.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of visibility data files for which flagging and calibration
information will be exported. Defaults to the list maintained in the
pipeline context.
default: []
example: vis=['X227.ms', 'X228.ms']
session -- List of sessions one per visibility file. Currently defaults
to a single virtual session containing all the visibility files in vis.
In future will default to set of observing sessions defined in the
context.

default: []
example: session=['session1', 'session2']
exportmses -- Export MeasurementSets defined in vis instead of flags,
caltables, and calibration instructions.
default: False
example: exportmses = True
pprfile -- Name of the pipeline processing request to be exported. Defaults
to a file matching the template 'PPR_*.xml'.
default: []
example: pprfile=['PPR_GRB021004.xml']
calintents -- List of calibrator image types to be exported. Defaults to
all standard calibrator intents 'BANDPASS', 'PHASE', 'FLUX'
default: ''
example: calintents='PHASE'
calimages -- List of calibrator images to be exported. Defaults to all
calibrator images recorded in the pipeline context.
default: []
example: calimages=['3C454.3.bandpass', '3C279.phase']
targetimages -- List of science target images to be exported. Defaults to all
science target images recorded in the pipeline context.
default: []
example: targetimages=['NGC3256.band3', 'NGC3256.band6']
products_dir -- Name of the data products subdirectory. Defaults to './'
default: ''
example: products_dir='../products'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hif_exportdata task exports the data defined in the pipeline context
and exports it to the data products directory, converting and or packing
it as necessary.
The current version of the task exports the following products
o an XML file containing the pipeline processing request
o a tar file per ASDM / MS containing the final flags version
o a text file per ASDM / MS containing the final calibration apply list
o a FITS image for each selected calibrator source image
o a FITS image for each selected science target source image
o a tar file per session containing the caltables for that session
o a tar file containing the file web log

o a text file containing the final list of CASA commands
Issues
Support for merging the calibration state information into the pipeline context / results structure and retrieving it still needs to be added.
Support for merging the clean results into the pipeline context / results structure and retrieving it still needs to be added.
Support for creating the final pipeline results entity still needs to be added.

Session information is not currently handled by the pipeline context.
By default all ASDMs are combined into one session.
Examples
1. Export the pipeline results for a single session to the data products directory

```
!mkdir ../products
hif_exportdata (products_dir='../products')
```

2. Export the pipeline results to the data products directory specify that only the gain calibrator images be saved.

```
!mkdir ../products
hif_exportdata (products_dir='../products', calintents='*PHASE*')
```

Parameter List

name	**type**	**default**	**description**
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
imaging_products_only	bool	False	Export the science target image products only
exportmses	bool	False	Export MeasurementSets instead of flags and caltables
pprfile	string	None	The pipeline processing request file to be exported
calintents	string	None	The calibrator source target intents to be exported
calimages	stringArray	None	List of calibrator images to be exported
targetimages	stringArray	None	List of target images to be exported
products_dir	string	None	The data products directory
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

h_import_calstate

Task Description

Import a calibration state from disk
Import a calibration state to disk.
Keyword arguments:
filename -- Name of the saved calibration state.
Description
h_import_calstate clears and then recreates the pipeline calibration state
based on the set of applycal calls given in the named file. The applycal
statements are interpreted in additive fashion; for identically specified
data selection targets, caltables specified in later statements will be added
to the state created by earlier calls.
Issues
Example
1. Import a calibration state from disk.
h_import_calstate(filename='aftergaincal.calstate')

Parameter List

name	type	default	description
filename	string	None	Name of the saved calibration state

Back

---

h_importdata

Task Description

Imports data into the interferometry pipeline
The h_importdata task loads the specified visibility data into the pipeline
context unpacking and / or converting it as necessary.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
session -- List of sessions to which the visibility files belong. Defaults
to a single session containing all the visibility files, otherwise
a session must be assigned to each vis file.
default: []

example: session=['session_1', 'session_2']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
asis -- ASDM tables to convert as is
default: 'Antenna Station Receiver CalAtmosphere'
example: 'Receiver', ''
process_caldevice -- Ingest the ASDM caldevice table
default: False
example: True
overwrite -- Overwrite existing MSs on output.
default: False
nocopy -- When importing an MS, disable copying of the MS to the working
directory
default: False
bdfflags -- Apply BDF flags on import
default: True
ocorr_mode -- Read in cross- and auto-correlation data(ca), cross-
correlation data only (co), or autocorrelation data only (ao).
default: ca
lazy -- Use the lazy import option
default: False
createmms -- Create a multi-MeasurementSet ('true') ready for parallel
processing, or a standard MeasurementSet ('false'). The default setting
('automatic') creates an MMS if running in a cluster environment.
default: automatic
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Load an ASDM list in the ../rawdata subdirectory into the context"
h_importdata(vis=['../rawdata/uid___A002_X30a93d_X43e',
'../rawdata/uid_A002_x30a93d_X44e'])
2. Load an MS in the current directory into the context:

h_importdata(vis=[uid___A002_X30a93d_X43e.ms])

3. Load a tarred ASDM in ../rawdata into the context:
   h_importdata(vis=['../rawdata/uid___A002_X30a93d_X43e.tar.gz'])
4. Check the h_importdata inputs, then import the data:
   myvislist = ['uid___A002_X30a93d_X43e.ms', 'uid_A002_x30a93d_X44e.ms']
   h_importdata(vis=myvislist, pipelinemode='getinputs')
   h_importdata(vis=myvislist)
5. Load an ASDM but check the results before accepting them into the context.
   results = h_importdata(vis=['uid___A002_X30a93d_X43e.ms'],
   acceptresults=False)
   results.accept()
6. Run in dryrun mode before running for real
   results = h_importdata(vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)
   results = h_importdata(vis=['uid___A002_X30a93d_X43e.ms'])

Parameter List

name	**type**	**default**	**description**
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of visibility data sessions
pipelinemode	string	automatic	The pipeline operating mode
asis	string	None	Extra ASDM tables to convert as is
process_caldevice	bool	False	Import the caldevice table from the ASDM
overwrite	bool	False	Overwrite existing files on import
nocopy	bool	False	Disable copying of MS to working directory
bdfflags	bool	True	Apply BDF flags on import
lazy	bool	False	Use the lazy import option
ocorr_mode	string	ca	ALMA default set to ca
createmms	string	automatic	Create an MMS
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

h_init

Task Description

Initialize the interferometry pipeline

The h_init task initializes the interferometry pipeline.

Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in 'interactive mode' or 'getinputs' modes

loglevel -- Pipeline log level threshold: (debug|info|warning|error|critical). Log messages below this threshold will not be displayed.
default: 'info'

plotlevel -- Pipeline plot level threshold: (all|default|summary). Toggle generation of detail plots in the web log. A level of 'all' generates all plots; 'summary' omits detail plots; 'default' generates all plots apart from for the hif_applycal task.
default: 'default'

output_dir -- Working directory for pipeline processing. Some pipeline processing products such as HTML logs and images will be directed to subdirectories of this path.
default: './' (current directory)

weblog -- Toggle web log generation.

overwrite -- Overwrite existing MSs on input.

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Description

h_init must be called before any other interferometry pipeline task. The pipeline can be initialized in one of two ways: by creating a new pipeline state (h_init) or be loading a saved pipeline state (h_resume).

h_init creates an empty pipeline context but does not load visibility data into the context. hif_importdata or hsd_importdata can be used to load data.

Issues
Examples
1. Create the pipeline context
h_init()

Parameter List

name	type	default	description
pipelinemode	string	automatic	The pipeline operating mode
loglevel	string	info	Log level for pipeline messages
plotlevel	string	default	Level for pipeline plots
output_dir	string	./	The output working directory
weblog	bool	True	Generate the web log
overwrite	bool	True	Overwrite existing files on import
dryrun	bool	False	Run the task (False) or display the task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

h_mssplit

Task Description

Select data from calibrated MS(s) to form new MS(s) for imaging
Create a list of science target MS(s) for imaging
Keyword Arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
--- pipeline parameter arguments which can be set in any pipeline mode
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets to be transformed. Defaults to the
list of MeasurementSets specified in the pipeline import data task.
default '': Split all MeasurementSets in the context.
example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
outputvis -- The list of output split MeasurementSets. The output list must
be the same length as the input list and the output names must be different

from the input names.
default '', The output name defaults to _split.ms
example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
field -- Select fields name(s) or id(s) to split.
default: '', All fields will be selected
example: '3C279', '"5795"' Note the double quotes around names which can be interpreted as numbers
intent -- Select intents to split
default: '', All data is selected.
example: 'TARGET'
spw -- Select spectral windows to split.
default: '', All spws are selected
example: '9', '9,13,15'
datacolumn -- Select spectral windows to split. The standard CASA options are supported
default: 'data',
example: 'corrected', 'model'
chanbin -- The channel binning factor. 1 for no binning, otherwise 2, 4, 8, or 16.
supported
default: 1,
example: 2, 4
timebin -- The time binning factor. '0s' for no binning
default: '0s'
example: '10s' for 10 second binning
replace -- If a split was performed delete the parent MS and remove it from the context.
default: True
example: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
Create new MeasurementSets for imaging from the corrected column of the input MeasurementSet. By default all science target data is copied to the new MS. The new MeasurementSet is not re-indexed to the selected data in the new MS will have the same source, field, and spw names and ids as it does in the parent MS.
Issues
TBD
Examples
1. Create a 4X channel smoothed output MS from the input MS
h_mssplit(chanbin=4)

Parameter List

name	type	default	description
vis	stringArray	None	The list of input MeasurementSets
outputvis	stringArray	None	The list of output split MeasurementSets
field	string	None	Set of data selection field names or ids, \'\' for all
intent	string	None	Set of data selection intents, \'\' for all
spw	string	None	Set of data selection spectral window ids \'\' for all
datacolumn	string	data	The data columns to process
chanbin	int	1	Channel bin width for spectral averaging
timebin	string	0s	Bin width for time averaging
replace	bool	True	Remove the parent MS and replace with the split MS
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

h_restoredata

Task Description

Restore flagged and calibration interferometry data from a pipeline run
The h_restoredata task restores flagged and calibrated MeasurementSets
from archived ASDMs and pipeline flagging and calibration date products.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of raw visibility data files to be restored. Assumed to be
in the directory specified by rawdata_dir.
default: None

example: vis=['uid___A002_X30a93d_X43e']
session -- List of sessions one per visibility file.
default: []
example: session=['session_3']
products_dir -- Name of the data products directory. Currently not
used.
default: '../products'
example: products_dir='myproductspath'
rawdata_dir -- Name of the rawdata subdirectory.
default: '../rawdata'
example: rawdata_dir='myrawdatapath'
lazy -- Use the lazy filler option
default: False
example: lazy=True
bdfflags -- Set the BDF flags
default: True
example: bdfflags=False
ocorr_mode -- Set ocorr_mode
default: 'ca'
example: ocorr_mode='ca'
asis -- Set list of tables to import as is
default: ''
example: ocorr_mode='Source Receiver'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The h_restoredata restores flagged and calibrated data from archived
ASDMs and pipeline flagging and calibration data products. Pending archive
retrieval support h_restoredata assumes that the required products
are available in the rawdata_dir in the format produced by the
h_exportdata task.
h_restoredata assumes that the following entities are available in the raw
data directory
o the ASDMs to be restored
o for each ASDM in the input list
o a compressed tar file of the final flagversions file, e.g.
uid___A002_X30a93d_X43e.ms.flagversions.tar.gz
o a text file containing the applycal instructions, e.g.
uid___A002_X30a93d_X43e.ms.calapply.txt

o a compressed tar file containing the caltables for the parent session,
e.g. uid___A001_X74_X29.session_3.caltables.tar.gz
h_restore data performs the following operations
o imports the ASDM(s))
o removes the default MS.flagversions directory created by the filler
o restores the final MS.flagversions directory stored by the pipeline
o restores the final set of pipeline flags to the MS
o restores the final calibration state of the MS
o restores the final calibration tables for each MS
o applies the calibration tables to each MS
Issues
Examples
1. Restore the pipeline results for a single ASDM in a single session
h_restoredata (vis=['uid___A002_X30a93d_X43e'], session=['session_1'], ocorr_mode='ca')

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
products_dir	string	../products	The archived pipeline data products directory
copytoraw	bool	True	Copy calibration and flagging tables to raw data directory
rawdata_dir	string	../rawdata	The rawdata directory
lazy	bool	False	Use the lazy filler option
bdfflags	bool	True	Set the BDF flags
ocorr_mode	string	ca	Correlation import mode
asis	string	None	List of tables to import asis
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

h_resume

Task Description

Restore a save pipeline state from disk
h_resume restores a name pipeline state from disk allowing a suspended
pipeline reduction session to be resumed.
Keyword parameters:
filename -- Name of the saved pipeline state. Setting filename to
'last' restores the most recently saved pipeline state whose
name begins with 'context*'.
default: 'last'
example: filename='context.s3.2012-02-13T10:49:11'
filename='last'
Description
h_resume restores a name pipeline state from disk allowing a suspended
pipeline reduction session to be resumed.
Issues
Examples
1. Resume the last saved session
h_resume()
2. Resume the named saved session
h_resume(filename='context.s3.2012-02-13T10:49:11')

Parameter List

name	type	default	description
filename	string	last	Filename of saved state to be restored

Back

---

h_save

Task Description

Save the pipeline state to disk
h_save saves the current pipeline state to disk under a unique name.
Keyword arguments:
filename -- Name of the saved pipeline state. If filename is '' then
a unique name will be generated computed from the root, 'context', the
current stage number, and the time stamp.
default: ''
Description
h_save saves the current pipeline state to disk under a unique name.
If no name is supplied one is generated automatically from a combination
of the root name, 'context', the current stage number, and a time stamp.

Issues
Example
1. Save the current state in the default file
   h_save()
2. Save the current state to a user named file
   h_save(filename='savestate_1')

Parameter List

name	type	default	description
filename	string	None	Name for saved state

Back

---

h_show_calstate

Task Description

Show the current pipeline calibration state
Keyword arguments:
None
Description
h_show_calstate displays the current on-the-fly calibration state of the pipeline as a set of equivalent applycal calls.

Parameter List

No parameter

Back

---

h_tsyscal

Task Description

Derive a Tsys calibration table
Derive the Tsys calibration for list of ALMA MeasurementSets.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
chantol -- The tolerance in channels for mapping atmospheric calibration windows (TDM) to science windows (FDM or TDM).

default: 1
example: chantol=5
---- pipeline parameter arguments which can be set in any pipeline mode
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- List of input visibility files.
default: none
example: vis=['ngc5921.ms']
caltable -- Name of output gain calibration tables.
default: none
example: caltable='ngc5921.gcal'
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
Derive the Tsys calibration for list of ALMA MeasurementSets.
Issues
Example

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltable(s)
chantol	int	1	Tsys spectral window map channel tolerance
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

h_weblog

Task Description

Open the pipeline weblog in a browser

Parameter List

No parameter

Back

Created at Fri Sep 14 10:04:26 2018 UTC
This page is automatically generated from XML interface definition files.

Summary of generic interferometric tasks and parameters

---

hif_antpos

Task Description

Derive an antenna position calibration table
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'
---- pipeline parameter arguments which can be set in any pipeline mode
hm_antpos -- Heuristics method for retrieving the antenna position
corrections. The options are 'online' (not yet implemented), 'manual',
and 'file'.
default: 'manual'
example: hm_antpos='file'
antenna -- The list of antennas for which the positions are to be corrected
if hm_antpos is 'manual'.
default: none
example: antenna='DV05,DV07'
offsets -- The list of antenna offsets for each antenna in 'antennas'. Each
offset is a set of 3 floating point numbers separated by commas, specified
in the ITRF frame.
default: none
example: offsets=[0.01, 0.02, 0.03, 0.03, 0.02, 0.01]
antposfile -- The file(s) containing the antenna offsets. Used if hm_antpos

is 'file'.
default: 'antennapos.csv'
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- List of input visibility files.
default: []
example: ['ngc5921.ms']
caltable -- Name of output gain calibration tables.
default: []
example: caltable=['ngc5921.gcal']
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
The hif_antpos task corrects the antenna positions recorded in the ASDMs using updated antenna position calibration information determined after the observation was taken.
Corrections can be input by hand, read from a file on disk, or in the future by querying an ALMA database service.
The antenna positions file is in 'csv' format containing 6 comma-delimited columns as shown below. The default name of this file is 'antennapos.csv'.
Contents of example 'antennapos.csv' file:
ms,antenna,xoffset,yoffset,zoffset,comment
uid___A002_X30a93d_X43e.ms,DV11,0.000,0.010,0.000,"No comment"
uid___A002_X30a93d_X43e.dup.ms,DV11,0.000,-0.010,0.000,"No comment"
The corrections are used to generate a calibration table which is recorded in the pipeline context and applied to the raw visibility data, on the fly to generate other calibration tables, or permanently to generate calibrated visibilities for imaging.
Issues
The hm_antpos 'online' option will be implemented when the observing system provides an antenna position determination service.
Example
1. Correct the position of antenna 5 for all the visibility files in a single pipeline run:
hif_antpos(antenna='DV05', offsets=[0.01, 0.02, 0.03])
2. Correct the position of antennas for all the visibility files in a single pipeline run using antenna positions files on disk. These files are assumed to conform to a default naming scheme if 'antposfile' is unspecified by the user:

hif_antpos(hm_antpos='file', antposfile='myantposfile.csv')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltable(s)
hm_antpos	string	manual	The antenna position determination method
antenna	string	None	List of antennas to be corrected
offsets	doubleArray	None	List of position corrections, one set per antenna
antposfile	string	None	File containing antenna position corrections
pipelinemode	string	automatic	The pipeline operation mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically accept results into context

Back

---

hif_applycal

Task Description

Apply the calibration(s) to the data
Apply precomputed calibrations to the data.
---- pipeline parameter arguments which can be set in any pipeline mode
applymode -- Calibration apply mode
''='calflagstrict': calibrate data and apply flags from solutions using
the strict flagging convention
'trial': report on flags from solutions, dataset entirely unchanged
'flagonly': apply flags from solutions only, data not calibrated
'calonly': calibrate data only, flags from solutions NOT applied
'calflagstrict':
'flagonlystrict': same as above except flag spws for which calibration is
unavailable in one or more tables (instead of allowing them to pass
uncalibrated and unflagged)
default: ''
flagsum -- Compute before and after flagging statistics summaries.
default: True
flagdetailedsum -- Compute detailed before and after flagging statistics summaries
if flagsum is True.
default: True

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
in the pipeline context.
default: []
example: ['X227.ms']
field -- A string containing the list of field names or field ids to which
the calibration will be applied. Defaults to all fields in the pipeline
context.
default: ''
example: '3C279', '3C279, M82'
intent -- A string containing the list of intents against which the
selected fields will be matched. Defaults to all supported intents
in the pipeline context.
default: ''
example: '*TARGET*'
spw -- The list of spectral windows and channels to which the calibration
will be applied. Defaults to all science windows in the pipeline
context.
default: ''
example: '17', '11, 15'
antenna -- The list of antennas to which the calibration will be applied.
Defaults to all antennas. Not currently supported.
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned
Description
hif_applycal applies the precomputed calibration tables stored in the pipeline
context to the set of visibility files using predetermined field and
spectral window maps and default values for the interpolation schemes.
Users can interact with the pipeline calibration state using the tasks
hif_export_calstate and hif_import_calstate.
Issues
There is some discussion about the appropriate values of calwt. Given

properly scaled data, the correct value should be the CASA default of True. However at the current time ALMA is suggesting that calwt be set to True for applying observatory calibrations, e.g. antenna positions, WVR, and system temperature corrections, and to False for applying instrument calibrations, e.g. bandpass, gain, and flux.
Examples
1. Apply the calibration to the target data
hif_applycal (intent='TARGET')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
applymode	string	None	Calibration mode: ""="calflagstrict","calflag","calflagstrict","trial","flagonly","flagonlystrict", or "calonly"
calwt	boolArray	True	Calibrate the weights as well as the data
flagbackup	bool	True	Backup the flags before the apply
flagsum	bool	True	Compute before and after flagging summary statistics
flagdetailedsum	bool	True	Compute detailed flagging statistics
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

hif_atmflag

Task Description

Flag channels with bad atmospheric transmission
hif_atmflag flags channels where the atmospheric transmission is low.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of

all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
intents -- Specifies the data intents whose channels are to be flagged if they have low atmospheric transmission. This string is inserted into the flagcmd given to the flagdata task applying the flags; it must have a valid flagcmd format.
default: '*AMP*,*BANDPASS*,*PHASE*'
flag_minabs -- True to flag channels with transmission less than fmin_limit.
default: False
fmin_limit -- The atmospheric transmission below which channels are to be flagged if flag_minabs is True.
default: 0.1
flag_nmedian -- True to flag channels based on transmission deviation from the median.
default: False
fnm_lo_limit -- Flag channels with transmission less than fnm_lo_limit * median transmission, if flag_nmedian is True.
default: 0.5
fnm_hi_limit -- Flag channels with transmission greater than fnm_hi_limit * median transmission, if flag_nmedian is True.
default: 1E9
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- List of input MeasurementSets.
default: [] - Use the MeasurementSets currently stored in the pipeline context.
example: vis=['X132.ms']
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- This parameter has no effect. The Tsyscal file is already in the pipeline context and is flagged in situ.
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
Spectral window channels with low atmospheric transmission are identified and flagged. The flagging view comprises a transmission spectrum for each spectral window calculated using the CASA atmosphere model.
Flags are generated by running the following rules on each spectrum:
If flag_minabs = True then channels with transmission below fmin_limit are flagged.
If flag_nmedian = True then channels with transmission below fnm_lo_limit * median transmission or above fnm_hi_limit * median transmission are flagged.
The flagging limits are set by frequency rather than by channel number.
The frequency frame is the native one of the spectral windows, usually TOPO.

Examples
1. Flag channels with transmission below 0.1 in each SpW:
   hif_atmflag()
equivalent to:
   hif_atmflag(flag_minabs=True, fmin_limit=0.1)
2. Flag channels with transmission below 0.4 * median transmission across the spectral window, and above 1.4 * median transmission:

   hif_atmflag(flag_nmedian=True, fnm_lo_limit=0.4, fnm_hi_limit=1.4)

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
intent	string	*AMP*,*BANDPASS*,*PHASE*	Data intents to which flags are to be applied
flag_minabs	bool	False	True to flag channels where transmission < fmin_limit
fmin_limit	double	0.1	Transmission limit below which channels are to be flagged
flag_nmedian	bool	False	True to flag channels based on deviation from median
fnm_lo_limit	double	0.5	If flag_nmedian then flag channels where transmission less than fnm_lo_limit * median transmission
fnm_hi_limit	double	1E9	If flag_nmedian then flag channels where transmission greater than fnm_hi_limit * median transmission
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

hif_bandpass

Task Description

Compute bandpass calibration solutions
Compute amplitude and phase as a function of frequency for each spectral window in each MeasurementSet.
Previous calibration can be applied on the fly.
Keyword arguments
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'
phaseup -- Do a phaseup on the data before computing the bandpass solution.
default: True
phaseupsolint -- The phase correction solution interval in CASA syntax.
Used when phaseup is True.
default: 'int'
example: phaseupsolint=300
phaseupbw -- Bandwidth to be used for phaseup. Defaults to 500MHz.
Used when phaseup is True.
default: ''
example: phaseupbw='' to use entire bandpass
phaseupbw='500MHz' to use central 500MHz
solint -- Time and channel solution intervals in CASA syntax.
default: 'inf,7.8125MHz'
example: solint='inf,10ch', 'inf'
combine -- Data axes to combine for solving. Axes are '', 'scan', 'spw', 'field' or any comma-separated combination.
default: 'scan'
example: combine='scan,field'
minblperant -- Minimum number of baselines required per antenna for each solve.
Antennas with fewer baselines are excluded from solutions.
default: 4
minsnr -- Solutions below this SNR are rejected.
default: 3.0
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context.
default: ''
example: vis=['M51.ms']
caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.

default: ''
example: caltable=['M51.bcal']
field -- The list of field names or field ids for which bandpasses are
computed. Defaults to all fields.
default: ''
example: field='3C279', field='3C279, M82'
intent -- A string containing a comma delimited list of intents against
which the selected fields are matched. Defaults to all data
with bandpass intent.
default: ''
example: intent='*PHASE*'
spw -- The list of spectral windows and channels for which bandpasses are
computed. Defaults to all science spectral windows.
default: ''
example: spw='11,13,15,17'
refant -- Reference antenna names. Defaults to the value(s) stored in the
pipeline context. If undefined in the pipeline context defaults to
the CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'
solnorm -- Normalise the bandpass solutions
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
hif_bandpass computes a bandpass solution for every specified science
spectral window. By default a 'phaseup' pre-calibration is performed
and applied on the fly to the data, before the bandpass is computed.
The hif_refant task may be used to precompute a prioritized list of
reference antennas.
Issues
There is currently some discussion about whether or not to do an 'ampup'
operations at the same time as the 'phaseup'. This is not required for the
bandpass computation but the amplitude information may provide a useful quality
assessment measure.
The specified minsnr parameter is currently applied to the bandpass
solution computation but not the 'phaseup' computation. Some noisy
solutions in the phaseup may not be properly rejected.
Examples

1. Compute a channel bandpass for all visibility files in the pipeline
context using the CASA reference antenna determination scheme:
   hif_bandpass()
2. Same as the above but precompute a prioritized reference antenna list:
   hif_refant()
   hif_bandpass()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna IDs
phaseup	bool	True	Phaseup before computing the bandpass
phaseupsolint	any	int	Phaseup correction solution interval
phaseupbw	string	None	Bandwidth to use for phaseup
solint	any	inf	Solution intervals
combine	string	scan	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	True	Normalise the bandpass solution
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

hif_checkproductsize

Task Description

Check imaging product size
Check interferometry imaging product size and try to mitigate to maximum allowed values. The task implements a mitigation cascade computing the largest cube size and trying to reduce it below a given limit by adjusting the nbins, hm_imsize and hm_cell parameters. If this step succeeds, it also checks the overall imaging product size and if necessary reduces the number of fields to be imaged.
Keyword arguments:
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata sets.
example: vis='ngc5921.ms'
vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
default: use all MeasurementSets in the context
maxcubesize -- Maximum allowed cube size mitigation goal in GB.
default: -1 (automatic from performance parameters)
example: 30.0
maxcubelimit -- Maximum allowed cube size mitigation failure limit in GB.
default: -1 (automatic from performance parameters)
example: 30.0
maxproductsize -- Maximum allowed product size mitigation goal and failure limit in GB.
default: -1 (automatic from performance parameters)
example: 200.0
calcsb -- Force (re-)calculation of sensitivities and beams
default=False
Options: False, True
parallel -- use multiple CPU nodes to compute dirty images
default: \'automatic\'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or

reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples:

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
maxcubesize	double	-1.0	Maximum cube size (mitigation goal)
maxcubelimit	double	-1.0	Maximum cube limit (mitigation failure limit)
maxproductsize	double	-1.0	Maximum product size (mitigation goal and failure limit)
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
parallel	string	automatic	Compute dirty images using MPI cluster
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hif_correctedampflag

Task Description

Flag corrected - model amplitudes based on calibrators.
Keyword arguments
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context.
default: ''

example: vis=['M51.ms']
field -- The list of field names or field ids for which bandpasses are
computed. If undefined (default), it will select all fields.
default: ''
example: field='3C279', '3C279, M82'
intent -- A string containing a comma delimited list of intents against
which the selected fields are matched. If undefined (default), it
will select all data with the BANDPASS intent.
default: ''
example: intent='*PHASE*'
spw -- The list of spectral windows and channels for which bandpasses are
computed. If undefined (default), it will select all science spectral
windows.
default: ''
example: spw='11,13,15,17'
antnegsig -- Lower sigma threshold for identifying outliers as a result of bad
antennas within individual timestamps.
default: 4.0
antpossig -- Upper sigma threshold for identifying outliers as a result of bad
antennas within individual timestamps.
default: 4.6
tmantint -- Threshold for maximum fraction of timestamps that are allowed to
contain outliers.
default: 0.063
tmint -- Initial threshold for maximum fraction of "outlier timestamps" over
"total timestamps" that a baseline may be a part of.
default: 0.085
tmbl -- Initial threshold for maximum fraction of "bad baselines" over "all
timestamps" that an antenna may be a part of.
default: 0.175
antblnegsig -- Lower sigma threshold for identifying outliers as a result of
"bad baselines" and/or "bad antennas" within baselines, across all
timestamps.
default: 3.4
antblpossig -- Upper sigma threshold for identifying outliers as a result of
"bad baselines" and/or "bad antennas" within baselines, across all
timestamps.
default: 3.2
relaxed_factor -- Relaxed value to set the threshold scaling factor to under
certain conditions (see task description).
default: 2.0
niter -- Maximum number of times to iterate on evaluation of flagging
heuristics. If an iteration results in no new flags, then subsequent
iterations are skipped.
default: 2
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but

do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Examples
1. Run default flagging on bandpass calibrator with recommended settings:
hif_correctedampflag()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
intent	string	None	Data intent to use in selecting data to evaluate for flagging
field	string	None	Field ids to use in selecting data to evaluate for flagging
spw	string	None	Spectral windows to use in selecting data to evaluate for flagging, \'\' for all
antnegsig	double	4.0	Lower sigma threshold for identifying outliers as a result of bad antennas within individual timestamps
antpossig	double	4.6	Upper sigma threshold for identifying outliers as a result of bad antennas within individual timestamps
tmantint	double	0.063	Threshold for maximum fraction of timestamps that are allowed to contain outliers
tmint	double	0.085	Initial threshold for maximum fraction of "outlier timestamps" over "total timestamps" that a baseline may be a part of
tmbl	double	0.175	Initial threshold for maximum fraction of "bad baselines" over "all timestamps" that an antenna may be a part of
antblnegsig	double	3.4	Lower sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines (across all timestamps)
antblpossig	double	3.2	Upper sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines (across all timestamps)
relaxed_factor	double	2.0	Relaxed value to set the threshold scaling factor to under certain conditions (see task description)
niter	int	2	Maximum number of times to iterate on evaluation of flagging heuristics.
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

hif_editimlist

Task Description

Add to a list of images to be produced with hif_makeimages()

Parameter List

name	type	default	description
imagename	string	None	Prefix for output image names.
search_radius_arcsec	double	1000.0	Size of the field finding beam search radius in arcsec.
cell	stringArray	None	Image X and Y cell size(s) with units or pixels per beam, \'\' for default. Single value same for both. \'ppb\' for pixels per beam.
conjbeams	bool	False	Use conjugate frequency in tclean for wideband A-terms.
cyclefactor	string	None	Controls the depth of clean in minor cycles based on PSF.
cycleniter	int	500	Controls max number of minor cycle iterations in a single major cycle.
deconvolver	string	None	Minor cycle algorithm (multiscale or mtmfs)
editmode	string	None	The edit mode of the task (e.g. add, edit)
field	stringArray	None	Set of data selection field names or ids, \'\' for all
imaging_mode	string	None	Identity of product type (e.g. VLASS quick look) desired. This will determine the heuristics used.
imsize	stringArray	None	Image X and Y size(s) in pixels or PB level (single fields), \'\' for default. Single value same for both. \'pb\' for PB level.
intent	string	None	Set of data selection intents
gridder	string	None	Name of the gridder to use with tclean
mask	string	None	Used to declare whether to use a predefined mask for tclean.
nbin	any	None	Channel binning factors per spw, \'\' for default.
nchan	string	None	Number of channels, -1 = all
niter	int	20000	Maximum number of clean iterations

nterms	int	2	None
parameter_file	string	None	None
phasecenter	any	None	Image center (direction or field index, \'\' for default)
reffreq	string	None	None
robust	double	1.0	Robustness parameter for Briggs weighting. robust = -2.0 maps to uniform weighting. robust = +2.0 maps to natural weighting. (robust=0.5 is equivalent to robust=0.0 in AIPS IMAGR.)
scales	string	None	The scales for multi-scale imaging.
specmode	string	None	Spectral gridding type (mfs, cont, cube, \'\' for default)
spw	string	None	Set of data selection spectral window/channels, \'\' for all
start	any	None	Channel start, \'\' for default
stokes	string	None	None
sensitivity	double	0.0	None
threshold	string	None	None
threshold_nsigma	double	4.0	None
uvtaper	string	None	Used to set a uv-taper during clean.
uvrange	string	None	Set of data selection uv ranges, \'\' for all.
width	any	None	Channel width, \'\' for default.
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hif_findcont

Task Description

Find continuum frequency ranges
Compute continuum ranges for all sources and spectral windows.
Keyword arguments:
--- pipeline parameter arguments which can be set in any pipeline mode

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In interactive mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
specified in the h_init or hif_importdata sets.
example: vis='ngc5921.ms'
vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
default: use all MeasurementSets in the context
target_list -- Dictionary specifying targets to be imaged; blank will read list from context
mosweight -- Mosaic weighting
default=None
Options: False, True
parallel -- use multiple CPU nodes to compute dirty images
default: \'automatic\'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples:

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
target_list	any	{}	Dictionary specifying targets to be imaged; blank will read list from context
mosweight	bool	None	Mosaic weighting
parallel	string	automatic	Compute dirty images using MPI cluster
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hif_gaincal

Task Description

Determine temporal gains from calibrator observations
Compute the gain solutions.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In interactive mode the user can set the pipeline context
defined parameters manually. In 'getinputs' mode the user can check the
settings of all pipeline parameters without running the task.
default: 'automatic'
hm_gtype -- The type of gain calibration. The options are 'gtype' and
'gspline' for CASA gain types = 'G' and 'GSPLINE' respectively.
calmode -- Type of solution. The options are 'ap' (amp and phase), 'p'
(phase only) and 'a' (amp only).
default: 'ap'
example: calmode='p', calmode='a', calmode='ap'
solint -- Time solution intervals in CASA syntax. Works for hm_gtype='gtype'
only.
default: 'inf'
example: solint='inf', solint='int', solint='100sec'
combine -- Data axes to combine for solving. Options are '','scan', 'spw',
'field' or any comma-separated combination. Works for hm_gtype='gtype'
only.
default: ''
example: combine=''
minblperant -- Minimum number of baselines required per antenna for each solve.

Antennas with fewer baselines are excluded from solutions. Works for hm_gtype='gtype' only.
default: 4
example: minblperant=2
minsnr -- Solutions below this SNR are rejected. Works for hm_gtype='channel' only.
default: 3.0
splinetime -- Spline timescale (sec). Used for hm_gtype='gspline'. Typical splinetime should cover about 3 to 5 calibrator scans.
default: 3600 (1 hour)
example: splinetime=1000
npointaver -- Tune phase-unwrapping algorithm. Used for hm_gtype='gspline'.
default: 3 (Keep at this value)
phasewrap -- Wrap the phase for changes larger than this amount (degrees). Used for hm_gtype='gspline'.
default: 180 (Keep at this value)
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context.
default: ''
example: vis=['M82A.ms', 'M82B.ms']
caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.
default: ''
example: caltable=['M82.gcal', 'M82B.gcal']
field -- The list of field names or field ids for which gain solutions are to be computed. Defaults to all fields with the standard intent.
default: ''
example: field='3C279', field='3C279, M82'
intent -- A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to *PHASE*.
default: ''
example: intent='', intent='*AMP*,*PHASE*'
spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all science spectral windows.
default: ''
example: spw='3C279', spw='3C279, M82'
smodel -- Point source Stokes parameters for source model (experimental) Defaults to using standard MODEL_DATA column data.
default: []
example: smodel=[1,0,0,0] - (I=1, unpolarized)
refant -- Reference antenna name(s) in priority order. Defaults to most recent values set in the pipeline context. If no reference antenna is defined in the pipeline context use the CASA defaults.
default: ''
example: refant='DV01', refant='DV05,DV07'

solnorm -- Normalise the gain solutions.
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The complex gains are derived from the data column (raw data) divided by the
model column (usually set with hif_setjy). The gains are obtained for a
specified solution interval, spw combination and field combination.
Good candidate reference antennas can be determined using the hif_refant
task.
Previous calibrations that have been stored in the pipeline context are
applied on the fly. Users can interact with these calibrations via the
hif_export_calstate and hif_import_calstate tasks.
Issues
The 'gspline' (smooth) option is still under development in CASA.
Examples
1. Compute standard per scan gain solutions that will be used to calibrate
the target:
hif_gaincal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
hm_gaintype	string	gtype	The gain solution type (gtype or gspline)
calmode	string	ap	Type of solution" (ap, p, a)
solint	any	inf	Solution intervals
combine	string	None	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	False	Normalize average solution amplitudes to 1.0
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
smodel	doubleArray	None	Point source Stokes parameters for source model
splinetime	double	3600.0	Spline timescale(sec)
npointaver	int	3	The phase-unwrapping algorithm
phasewrap	double	180.0	Wrap the phase for jumps greater than this value (degrees)
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

hif_gainflag

Task Description

Flag antennas with deviant gain
hif_gainflag flags data for antennas with deviant median gains and/or high gain rms.
Keyword arguments

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
flag_mediandeviant -- True to flag antennas with deviant median gain,
calculated as:

abs(median(antenna) - median(all antennas)) / MAD(all antennas)
> fmeddev_limit

default: False
fmeddev_limit -- Flag antennas with \'mediandeviant\' metric larger than
fmeddev_limit.
default: 3.0
flag_rmsdeviant -- True to flag antennas with deviant gain rms, calculated as:
stdev(antenna) / MAD(all antennas) > frmsdev_limit
default: False
frmsdev_limit -- Flag antennas with 'rmsdeviant' metric larger than
frmsdev_limit.
default: 8.0
flag_nrmsdeviant -- True to flag antennas with deviant normalised gain rms,
calculated as:
deviation (ant_i) = (sm(ant_i) - med_sm_allant) / sigma_sm_allant

where:
sm(ant_i) = sigma(ant_i) / median(ant_i)
sigma_sm_allant = 1.4826 * mad({sm(ant_1), sm(ant_2), .., sm(ant_nants)})
med_sm_allant = median({sm(ant_1), sm(ant_2), ..., sm(ant_nants)})
default: True
fnrmsdev_limit -- Flag antennas with 'nrmsdeviant' metric larger than
fnrmsdev_limit.
default: 6.0
metric_order -- Order in which the flagging metrics are evaluated.
default: 'mediandeviant, rmsdeviant, nrmsdeviant'
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- List of input MeasurementSets.
default: [] - Use the MeasurementSets currently known to the pipeline
context.
intent -- A string containing the list of intents to be checked for antennas
with deviant gains. The default is blank, which causes the task to select
the 'BANDPASS' intent.
default: ''
example: intent='*BANDPASS*'

spw -- The list of spectral windows and channels to which the calibration
    will be applied. Defaults to all science windows in the pipeline
    context.
    default: ''
    example: spw='17', spw='11, 15'
refant -- A string containing a prioritized list of reference antenna name(s)
    to be used to produce the gain table. Defaults to the value(s) stored in
    the pipeline context. If undefined in the pipeline context defaults to the
    CASA reference antenna naming scheme.
    default: ''
    example: refant='DV01', refant='DV06,DV07'
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
    do not execute (False).
    default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
    reject them (False).
    default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
    the results object for the pipeline task is returned.
Description
Deviant antennas are detected by analysis of a view showing their calibration gains. Three flagging metrics are available (mediandeviant, rmsdeviant, nrmsdeviant) and a flagging view is created for each metric that has been enabled. Each view is a list of 2D images with axes 'Time' and 'Antenna'; there is one image for each spectral window and intent. If any of the flagging metrics exceeds their corresponding limit (fmeddev_limit, frmsdev_limit, fnrmsdev_limit) for a given antenna in a given spw, then all data for that antenna and for all spws within the same baseband will be flagged with a flagcmd.
The following heuristics are used for the flagging metrics:
  mediandeviant:
    abs(median(antenna) - median(all antennas)) / MAD(all antennas)
      > fmeddev_limit
  rmsdeviant:
    stdev(antenna) / MAD(all antennas) > frmsdev_limit
  nrmsdeviant:
    deviation (ant_i) = (sm(ant_i) - med_sm_allant) / sigma_sm_allant
    with:
     sm(ant_i) = sigma(ant_i) / median(ant_i)
     sigma_sm_allant = 1.4826 * mad({sm(ant_1), sm(ant_2), .., sm(ant_nants)})
     med_sm_allant = median({sm(ant_1), sm(ant_2), ..., sm(ant_nants)})

  where MAD is the median absolute deviation from the median.

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
intent	string	None	Data intent whose gains are to checked
spw	string	None	Spectral window ids whose gains are to be checked
refant	string	None	Reference antenna names
flag_mediandeviant	bool	False	True to flag antennas with deviant median gains
fmeddev_limit	double	3.0	Flag antennas with \'mediandeviant\' metric above fmeddev_limit
flag_rmsdeviant	bool	False	True to flag antennas with deviant gain rms
frmsdev_limit	double	8.0	Flag antennas with \'rmsdeviant\' metric above frmsdev_limit
flag_nrmsdeviant	bool	True	True to flag antennas with deviant normalised gain rms
fnrmsdev_limit	double	6.0	Flag antennas with \'nrmsdeviant\' metric above fnrmsdev_limit
metric_order	string	mediandeviant, rmsdeviant, nrmsdeviant	Order in which the flagging metrics are evaluated
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

---

hif_linfeedpolcal

Task Description

Base linfeedpolcal task
The hif_linfeedpolcal task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode

vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic linfeedpolcal task
hif_linfeedpolcal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hif_lowgainflag

Task Description

Flag antennas with low or high gain
hif_lowgainflag flags data for antennas with unusually low or high gains.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
flag_nmedian -- True to flag figures of merit greater than
fnm_hi_limit * median or lower than fnm_lo_limit * median.
default: True
fnm_lo_limit -- Points lower than fnm_lo_limit * median are flagged.
default: 0.5
fnm_hi_limit -- Points greater than fnm_hi_limit * median are flagged.
default: 1.5
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- List of input MeasurementSets.
default: [] - Use the MeasurementSets currently known to the pipeline
context.
intent -- A string containing the list of intents to be checked for antennas
with deviant gains. The default is blank, which causes the task to select
the 'BANDPASS' intent.
default: ''
example: intent='*BANDPASS*'
spw -- The list of spectral windows and channels to which the calibration will
be applied. Defaults to all science windows in the pipeline context.
default: ''
example: spw='17', spw='11, 15'
refant -- A string containing a prioritized list of reference antenna name(s)
to be used to produce the gain table. Defaults to the value(s) stored in
the pipeline context. If undefined in the pipeline context defaults to the
CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).

default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
Deviant antennas are detected by analysis of a view showing their calibration gains. This view is a list of 2D images with axes 'Time' and 'Antenna'; there is one image for each spectral window and intent. A flagcmd to flag all data for an antenna will be generated by any gain that is outside the range [fnm_lo_limit * median, fnm_hi_limit * median].

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
intent	string	None	Data intent whose gains are to checked
spw	string	None	Spectral window ids whose gains are to be checked
refant	string	None	Reference antenna names
flag_nmedian	bool	True	True to flag values outside range [fnm_lo_limit * median, fnm_hi_limit*nmedian]
fnm_lo_limit	double	0.5	Flag values lower than fnm_lo_limit * median
fnm_hi_limit	double	1.5	Flag values higher than fnm_hi_limit * median
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

---

hif_makecutoutimages

Task Description

Base makecutoutimages task
The hif_makecutoutimages task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs, MSs, or tar files of MSs, If ASDM files are specified, they will be converted to MS format.
default: []

example: vis=['X227.ms', 'asdms.tar.gz']
offsetblc -- -x and -y offsets to the bottom lower corner (blc) in arcseconds
default: [0.0,0.0]
offsettrc -- +x and +y offsets to the top right corner (trc) in arcseconds
default: [0.0,0.0]
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic makecutoutimages task
hif_makecutoutimages()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
offsetblc	doubleArray	None	-x and -y offsets to the bottom lower corner (blc) in arcseconds
offsettrc	doubleArray	None	+x and +y offsets to the top right corner (trc) in arcseconds
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hif_makeimages

Task Description

Compute clean map
Compute a cleaned image for a particular target source/intent and spectral window.
Keyword arguments:
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata sets.
example: vis='ngc5921.ms'
vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
default: use all MeasurementSets in the context
hm_masking -- Clean masking mode. Options are 'centralregion', 'psf', 'psfiter', 'auto', 'manual' and 'none'
default: 'auto'
example: 'manual'
hm_sidelobethreshold -- sidelobethreshold * the max sidelobe level
hm_noisethreshold -- noisethreshold * rms in residual image
hm_lownoisethreshold -- lownoisethreshold * rms in residual image
hm_negativethreshold -- negativethreshold * rms in residual image
hm_minbeamfrac -- negativethreshold * rms in residual image
hm_growiterations -- Number of binary dilation iterations for growing the mask
hm_dogrowprune -- Do pruning on the grow mask
hm_minpercentchange -- Mask size change threshold
cleancontranges -- Clean continuum frequency ranges in cubes
default=False
Options: False, True
calcsb -- Force (re-)calculation of sensitivities and beams
default=False
Options: False, True
mosweight -- Mosaic weighting
default=None
Options: False, True
parallel -- use multiple CPU nodes to clean images
default: \'automatic\'

--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
  do not execute (False).
  default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
  reject them (False).
  default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
   the results object for the pipeline task is returned.
Examples:

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
target_list	any	{}	Dictionary specifying targets to be imaged; blank will read list from context
hm_masking	string	auto	Pipeline heuristics masking option
hm_sidelobethreshold	double	-999.0	sidelobethreshold * the max sidelobe level
hm_noisethreshold	double	-999.0	noisethreshold * rms in residual image
hm_lownoisethreshold	double	-999.0	lownoisethreshold * rms in residual image
hm_negativethreshold	double	-999.0	negativethreshold * rms in residual image
hm_minbeamfrac	double	-999.0	Minimum beam fraction for pruning
hm_growiterations	int	-999	Number of binary dilation iterations for growing the mask
hm_dogrowprune	bool	True	Do pruning on the grow mask
hm_minpercentchange	double	-999.0	Mask size change threshold
hm_cleaning	string	None	Pipeline cleaning mode
tlimit	double	2.0	Times the sensitivity limit for cleaning
masklimit	int	4	Times good mask pixels for cleaning
maxncleans	int	1	Maximum number of clean task calls
cleancontranges	bool	False	Clean continuum frequency ranges in cubes
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
mosweight	bool	None	Mosaic weighting
parallel	string	automatic	Clean images using MPI cluster
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

hif_makeimlist

Task Description

Compute list of clean images to be produced
Create a list of images to be cleaned.
Keyword Arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
--- pipeline parameter arguments which can be set in any pipeline mode
specmode -- Frequency imaging mode, 'mfs', 'cont', 'cube', 'repBW'. \'\' defaults to
'cube' if intent parameter includes 'TARGET' otherwise 'mfs'.
default: ''
example: 'mfs', 'cont', 'cube', 'repBW'
example:
specmode='mfs' produce one image per source and spw
specmode='cont' produce one image per source and aggregate over all
specified spws
specmode='cube' produce an LSRK frequency cube, channels are specified
in frequency
specmode='repBW' produce an LSRK frequency cube at representative channel width
hm_cell -- Cell size (x, y)
default '' Compute cell size based on the UV coverage of all the fields
to be imaged and use a 5 pix per beam sampling.
The pix per beam specification uses the above default cell size
('5ppb') and scales it accordingly.
example: ['0.5arcsec', '0.5arcsec'] '3ppb'
hm_imsize -- Image X and Y size in pixels or PB level for single field. The
explicit sizes must be even and divisible by 2,3,5,7 only.
default: '' The default values are derived as follows:
1. Determine phase center and spread of field centers around it.
2. Set the size of the image to cover the spread of field centers plus
a border of width 0.75 * beam radius, to first null.
3. Divide X and Y extents by cell size to arrive at the number of
pixels required.
The PB level setting for single fields leads to an imsize extending
to the specified level plus 5% padding in all directions.
example: [120, 120], '0.3pb'
calmaxpix -- Maximum image X or Y size in pixels if a calibrator is being
imaged ('PHASE', 'BANDPASS', 'AMPLITUDE' or 'FLUX').
default: 300
example: 300
width -- Output channel width.

default: '' Difference in frequency between first 2 selected channels.
for frequency mode images.
example: '24.2kHz'
'pilotimage' for 15 MHz / 8 channel heuristic
nbins -- Channel binning factors per spw
default: '' Binning factors for each spw. Format: 'spw1:nb1,spw2:nb2,...'
Optional wildcard: '*:nb'
example: '9:2,11:4,13:2,15:8'
'*:2'
robust -- Briggs robustness parameter
default=-999.0; example: robust=0.7;
Options: -2.0 to 2.0; -2 (uniform)/+2 (natural)
uvtaper -- uv-taper on outer baselines
default=[]
example: ['10arcsec']
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
specified in the h_init or hif_importdata sets.
default '': use all MeasurementSets in the context
example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
intent -- Select intents for which associated fields will be imaged.
default: 'TARGET'
example: 'PHASE,BANDPASS'
field -- Select fields to image. Use field name(s) NOT id(s). Mosaics
are assumed to have common source / field names. If intent is specified
only fields with data matching the intent will be selected. The fields
will be selected from MeasurementSets in 'vis'.
default: '' Fields matching intent, one image per target source.
example: '3C279', 'Centaurus*', '3C279,J1427-421'
spw -- Select spectral window/channels to image.
default: '' Individual images will be computed for all science spectral
windows.
example: '9'
contfile -- Name of file with frequency ranges to use for continuum images.
default: 'cont.dat'
example: 'mycont.dat'
linesfile -- Name of file with line frequency ranges to exclude for continuum images.
default: 'lines.dat'
example: 'mylines.dat'
uvrange -- Select a set of uv ranges to image.
default: '' All uv data is included
example: '0~1000klambda', ['0~100klambda', 100~1000klambda]
phasecenter -- Direction measure or field id of the image center.
default: '' The default phase center is set to the mean of the field
directions of all fields that are to be image together.
example: 0, 'J2000 19h30m00 -40d00m00'

nchan -- Total number of channels in the output image(s)
    default: -1 Selects enough channels to cover the data selected by
        spw consistent with start and width.
    example: 100
start -- First channel for frequency mode images.
    default '' Starts at first input channel of the spw.
    example: '22.3GHz'
clearlist -- Clear previous image target list.
    default: True
    example: False
per_eb -- Make an image target per EB.
    default False
    example: True
calcsb -- Force (re-)calculation of sensitivities and beams
            default=False
            Options: False, True
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
   do not execute (False).
   default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
   reject them (False).
   default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
    the results object for the pipeline task is returned.
Description
Generate a list of images to be cleaned. By default the list will include
one image per science target per spw. Calibrator targets can be selected
by setting appropriate values for intent.
By default the output image cell size is set to the minimum cell size
consistent with the UV coverage.
By default the image size in pixels is set to values determined by the
cell size and the single dish beam size. If a calibrator is being
imaged (intents 'PHASE', 'BANDPASS', 'FLUX' or 'AMPLITUDE') then the
image dimensions are limited to 'calmaxpix' pixels.
By default science target images are cubes and calibrator target images
are single channel. Science target images may be mosaics or single fields.
Issues
TBD
Examples
1. Make a list of science target images to be cleaned, one image per science
spw.
    hif_makeimlist()
2. Make a list of PHASE and BANDPASS calibrator targets to be imaged,
one image per science spw.
    hif_makeimlist(intent='PHASE,BANDPASS')

3. Make a list of PHASE calibrator images observed in spw 1, images limited to 50 pixels on a side.

```
hif_makeimlist(intent='PHASE',spw='1',calmaxpix=50)
```

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
imagename	string	None	Prefix for output image names, \'\' for default.
intent	string	TARGET	Set of data selection intents
field	string	None	Set of data selection field names or ids, \'\' for all
spw	string	None	Set of data selection spectral window/channels, \'\' for all
contfile	string	None	Name of file with frequency ranges to use for continuum images.
linesfile	string	None	Name of file with line frequency ranges to exclude for continuum images.
uvrange	string	None	Set of data selection uv ranges, \'\' for all.
specmode	string	None	Spectral gridding type (mfs, cont, cube, repBW, \'\' for default)
outframe	string	None	velocity frame of output image (LSRK, \'\' for default)
hm_imsize	stringArray	None	Image X and Y size(s) in pixels or PB level (single fields), \'\' for default. Single value same for both. \'pb\' for PB level.
hm_cell	stringArray	None	Image X and Y cell size(s) with units or pixels per beam, \'\' for default. Single value same for both. \'ppb\' for pixels per beam.
calmaxpix	int	300	Maximum X and Y size of calibrator images in pixels
phasecenter	any	None	Image center (direction or field index, \'\' for default)
nchan	int	-1	Number of channels, -1 = all
start	any	None	Channel start, \'\' for default
width	any	None	Channel width, \'\' for default.
nbins	any	None	Channel binning factors per spw, \'\' for default.
robust	double	-999.0	Briggs robustness parameter
uvtaper	stringArray	None	uv-taper on outer baselines
clearlist	bool	True	Clear any previous target list

per_eb	bool	False	Make an image target per EB
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hif_makermsimages

Task Description

Base makermsimages task
The hif_makermsimages task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic makermsimages task
hif_makermsimages()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hif_mstransform

Task Description

Select data from calibrated MS(s) to form new MS(s) for imaging
Create a list of science target MS(s) for imaging
Keyword Arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
--- pipeline parameter arguments which can be set in any pipeline mode
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets to be transformed. Defaults to the
list of MeasurementSets specified in the pipeline import data task.
default '': Transform all calibration MeasurementSets in the context.
example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
outputvis -- The list of output transformed MeasurementSets to be used for imaging.
The output list must be the same length as the input list.
default '', The output name defaults to _target.ms
example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
field -- Select fields name(s) or id(s) to transform. Only fields with data matching
the intent will be selected.
default: '', Fields matching intent.
example: '3C279', 'Centaurus*', '3C279,J1427-421'
intent -- Select intents for which associated fields will be imaged.
default: '', Only TARGET data is selected.
example: 'PHASE,BANDPASS'
spw -- Select spectral window/channels to image.
default: '', All science spws for which the specified intent is valid are selected

example: ’9’
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is ’getinputs’ then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
Create new MeasurementSets for imaging from the corrected column of the input
MeasurementSet. By default all science target data is copied to the new MS. The new
MeasurementSet is not re-indexed to the selected data and the new MS will have the
same source, field, and spw names and ids as it does in the parent MS.
Issues
TBD
Examples
1. Create a science target MS from the corrected column in the input MS.
hif_mstransform()
2. Make a phase and bandpass calibrator targets MS from the corrected
column in the input MS.
hif_mstransform(intent=’PHASE,BANDPASS’)

Parameter List

name	type	default	description
vis	stringArray	None	The list of input MeasurementSets
outputvis	stringArray	None	The list of transformed MeasurementSets to be used for imaging
field	string	None	Set of data selection field names or ids, \’\’ for all
intent	string	None	Set of data selection intents
spw	string	None	Set of data selection spectral window ids \’\’ for all
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

hif_polarization

Task Description

Base polarization task
The hif_polarization task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic polarization task
hif_polarization()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

hif_rawflagchans

Task Description

Flag deviant baseline/channels in raw data
hif_rawflagchans flags deviant baseline/channels in the raw data.
The flagging views used are derived from the raw data for the specified
intent - default is BANDPASS.
Bad baseline/channels are flagged for all intents, not just the
one that is the basis of the flagging views.
For each spectral window the flagging view is a 2d image with axes
'channel' and 'baseline'. The pixel for each channel, baseline is the
time average of the underlying unflagged raw data.
The baseline axis is labeled by numbers of form id1.id2 where id1 and id2
are the IDs of the baseline antennas. Both id1 and id2 run over all
antenna IDs in the observation. This means that each baseline is shown
twice but has the benefit that 'bad' antennas are easily identified by
eye.
Three flagging methods are available:
If parameter flag_hilo is set True then outliers from the median
of each flagging view will be flagged.
If parameter flag_bad_quadrant is set True then a simple 2 part
test is used to check for bad antenna quadrants and/or bad baseline
quadrants. Here a 'quadrant' is defined simply as one quarter of the
channel axis. The first part of the test is to note as 'suspect' those
points further from the view median than fbq_hilo_limit * MAD.
The second part is to flag entire antenna/quadrants if their
fraction of suspect points exceeds fbq_antenna_frac_limit.
Failing that, entire baseline/quadrants may be flagged if their
fraction of suspect points exceeds fbq_baseline_frac_limit.
Suspect points are not flagged unless as part of a bad antenna or
baseline quadrant.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
flag_hilo -- True to flag channel/baseline data further from the view
median than fhl_limit * MAD.
default: True
fhl_limit -- If flag_hilo is True then flag channel/baseline data
further from the view median than fhl_limit * MAD.

default: 20
fhl_minsample -- Do no flagging if the view median and MAD are derived
from fewer than fhl_minsample view pixels.
default: 5
flag_bad_quadrant -- True to search for and flag bad antenna quadrants
and baseline quadrants. Here a /'quadrant/' is one
quarter of the channel axis.
default: True
fbq_hilo_limit -- If flag_bad_quadrant is True then channel/baselines further
from the view median than fbq_hilo_limit * MAD will be noted as
'suspect'. If there are enough of them to indicate that an antenna or
baseline quadrant is bad then all channel/baselines in that quadrant will
be flagged.
default: 8.0
fbq_antenna_frac_limit -- If flag_bad_quadrant is True and the fraction of
suspect channel/baselines in a particular antenna/quadrant exceeds
fbq_antenna_frac_limit then all data for that antenna/quadrant will
be flagged.
default: 0.2
fbq_baseline_frac_limit -- If flag_bad_quadrant is True and the fraction of
suspect channel/baselines in a particular baseline/quadrant exceeds
fbq_baseline_frac_limit then all data for that baseline/quadrant will
be flagged.
default: 1.0 (i.e. no flagging)
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- List of input MeasurementSets.
default: [] - Use the MeasurementSets currently known to the pipeline
context.
intent -- A string containing the list of intents to be checked for antennas
with deviant gains. The default is blank, which causes the task to select
the 'BANDPASS' intent.
default: ''
example: intent='*BANDPASS*'
spw -- The list of spectral windows and channels to which the calibration
will be applied. Defaults to all science windows in the pipeline
context.
default: ''
example: spw='17', spw='11, 15'
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- This parameter has no effect. The Tsyscal file is already
in the pipeline context and is flagged in situ.
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise

the results object for the pipeline task is returned.

Examples

1. Flag bad quadrants and wild outliers, default method:

hif_rawflagchans()

equivalent to:

hif_rawflagchans(flag_hilo=True, fhl_limit=20,
flag_bad_quadrant=True, fbq_hilo_limit=8,
fbq_antenna_frac_limit=0.2, fbq_baseline_frac_limit=1.0)

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
spw	string	None	Set of data selection spectral windows, \'\' for all
intent	string	None	Data intent to use in creating flagging view
flag_hilo	bool	True	True to flag outlier baseline/channels
fhl_limit	double	20.0	Flag baseline/channels further from median than limit * MAD
fhl_minsample	double	5	Minimum number of points in sample
flag_bad_quadrant	bool	True	True to flag bad quadrants in antennas
fbq_hilo_limit	double	8.0	Note as \'suspect\' baseline/channels further from median than limit * MAD
fbq_antenna_frac_limit	double	0.2	Flag antenna quadrant if its fraction of \'suspect\' baseline/channels > limit
fbq_baseline_frac_limit	double	1.0	Flag baseline quadrant if its fraction of \'suspect\' baseline/channels > limit
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

hif_refant

Task Description

Select the best reference antennas
The hif_refant task selects a list of reference antennas and outputs them
in priority order. The priority order is determined by a weighted combination
of scores derived by the antenna selection heuristics.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.
hm_refant -- The heuristics method or mode for selection the reference
antenna. The options are 'manual' and 'automatic. In manual
mode a user supplied reference antenna refant is supplied.
In 'automatic' mode the antennas are selected automatically.
default: 'automatic'
refant -- The user supplied reference antenna for 'manual' mode. If
no antenna list is supplied an empty list is returned.
default: ''
example: 'DV05'
geometry -- Score antenna by proximity to the center of the array. This
option is quick as only the ANTENNA table must be read.
default: True
flagging -- Score antennas by percentage of unflagged data. This option
requires computing flagging statistics.
default: True
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
in the pipeline context.
default: ''
example: ['M31.ms']
field -- The comma delimited list of field names or field ids for which flagging
scores are computed if hm_refant='automatic' and flagging = True
default: '' (Default to fields with the specified intents)
example: '3C279', '3C279,M82'
intent -- A string containing a comma delimited list of intents against
which the selected fields are matched. Defaults to all supported
intents.
default: 'AMPLI,BANDPASS,PHASE'
example: 'BANDPASS'
spw -- A string containing the comma delimited list of spectral window ids for

which flagging scores are computed if hm_refant='automatic' and flagging = True.
default: '' (all spws observed with the specified intents)
example: '11,13,15,17'
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hif_refant task selects a list of reference antennas and stores them
in the pipeline context in priority order.
The priority order is determined by a weighted combination of scores derived
by the antenna selection heuristics. In manual mode the reference antennas
can be set by hand.
Issues
Examples
1. Compute the references antennas to be used for bandpass and gain calibration.
hif_refant()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	List of field names or ids
spw	string	None	List of spectral windows ids
intent	string	AMPLITUDE,BANDPASS,PHASE	List of data selection intents
hm_refant	string	automatic	The reference antenna heuristics mode
refant	string	None	List of reference antennas
geometry	bool	True	Score by proximity to center of the array
flagging	bool	True	Score by percentage of good data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results into the pipeline context
refantignore	string	None	String list of antennas to ignore

Back

---

hif_setjy

Task Description

Fill the model column with calibrated visibilities
Fills the model column with the model visibilities.
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
fluxdensity -- Specified flux density [I,Q,U,V] in Jy. Uses [1,0,0,0]
flux density for unrecognized sources, and standard flux densities for
ones recognized by 'standard', including 3C286, 3C48, 3C147, and several
planets, moons, and asteroids.
default: -1

example: [3.06,0.0,0.0,0.0]
reffile -- Path to a file containing flux densities for calibrators unknown to CASA. Values given in this file take precedence over the CASA-derived values for all calibrators except solar system calibrators. By default the path is set to the CSV file created by h_importdata, consisting of catalogue fluxes extracted from the ASDM.
default: ''
example: reffile='', reffile='working/flux.csv'
spix -- Spectral index for fluxdensity S = fluxdensity * (freq/reffreq)**spix
Only used if fluxdensity is being used. If fluxdensity is positive, and spix is nonzero, then reffreq must be set too. It is applied in the same way to all polarizations, and does not account for Faraday rotation or depolarization.
default: 0
reffreq -- The reference frequency for spix, given with units. Provided to avoid division by zero. If the flux density is being scaled by spectral index, then reffreq must be set to whatever reference frequency is correct for the given fluxdensity and spix. It cannot be determined from vis. On the other hand, if spix is 0, then any positive frequency can be used and will be ignored.
default: '1GHz'
example: reffreq='86.0GHz', reffreq='4.65e9Hz'
scalebychan -- This determines whether the fluxdensity set in the model is calculated on a per channel basis. If False then only one fluxdensity value is calculated per spw.
default: True
standard -- Flux density standard, used if fluxdensity[0] less than 0.0. The options are: 'Baars','Perley 90','Perley-Taylor 95', 'Perley-Taylor 99', 'Perley-Butler 2010' and 'Butler-JPL-Horizons 2010'.
default: 'Butler-JPL-Horizons 2012' for solar system object
'Perley-Butler 2010' otherwise
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets defined in the pipeline context.
default: []
field -- The list of field names or field ids for which the models are to be set. Defaults to all fields with intent '*AMPLITUDE*'.
default: ''
example: field='3C279', field='3C279, M82'
intent -- A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to all data with amplitude intent.
default: ''
example: intent='*AMPLITUDE*'
spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.

default: ''
example: spw='11,13,15,17'
model -- Model image for setting model visibilities. Not fully
supported.
default: ''
example: see details in help for CASA setjy task
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
Fills the model column with the model visibilities.
Issues
Support for the setjy spix parameter needs to be added.
Examples
1. Set the model flux densities for all the amplitude calibrators:
hif_setjy()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	List of field names or ids
intent	string	None	Observing intent of flux calibrators
spw	string	None	List of spectral window ids
model	string	None	File location for field model
reffile	string	None	Path to file with fluxes for non-solar system calibrators
normfluxes	bool	False	Normalize lookup fluxes
reffreq	string	1GHz	Reference frequency for spix
fluxdensity	any	-1	Specified flux density [I,Q,U,V]; -1 will lookup values
spix	double	0.0	Spectral index of fluxdensity
scalebychan	bool	True	Scale the flux density on a per channel basis or else on a per spw basis
standard	variant	None	Flux density standard
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the commands (True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

hif_setmodels

Task Description

Set calibrator source models
Derive flux densities for point source transfer calibrators using flux models
for reference calibrators.
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the users can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
reffile -- The reference file containing a lookup table of point source models
This file currently defaults to 'flux.csv' in the working directory. This

file must conform to the standard pipeline 'flux.csv' format
default: ''
example: 'myfluxes.csv'
normfluxes -- Normalize the transfer field lookup fluxes to 1.0. The reference
field fluxes are never normalized.
default: True
example: False
scalebychan -- Scale the flux density on a per channel basis or else on a per spw basis
default: True
example: False
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context
default: ''
example: ['M32A.ms', 'M32B.ms']
reference -- A string containing a comma delimited list of field names
defining the reference calibrators. Defaults to field names with
intent 'AMPLITUDE'.
default: ''
example: 'M82,3C273'
refintent -- A string containing a comma delimited list of intents
used to select the reference calibrators. Defaults to 'AMPLITUDE'.
default: 'AMPLITUDE'. '' Means no sources.
example: 'BANDPASS'
transfer -- A string containing a comma delimited list of field names
defining the transfer calibrators. Defaults to field names with
intent ''.
default: 'BANDPASS,PHASE,CHECK'
example: 'J1328+041,J1206+30'
transintent -- A string containing a comma delimited list of intents
defining the transfer calibrators. Defaults to 'BANDPASS,PHASE,CHECK'.
'' stands for no transfer sources.
default: 'BANDPASS,PHASE,CHECK'
example: 'PHASE'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned
Description
Set model fluxes values for calibrator reference and transfer sources using lookup

values. By default the reference sources are the flux calibrators and the transfer sources are the bandpass, phase, and check source calibrators. Reference sources which are also in the transfer source list are removed from the transfer source list.
Built-in lookup tables are used to compute models for solar system object calibrators.
Point source models for other calibrators are provided in the reference file.
Normalize fluxes are computed for transfer sources if the normfluxes parameter is set to True.
The reference file default to a file called 'flux.csv' in the current working directory. This file is normal created in the importdata step. The file is in 'csv' format and contains the following comma delimited columns.
vis,fieldid,spwid,I,Q,U,V,pix,comment
Issues
Examples
1. Set model fluxes for the flux, bandpass, phase, and check sources.
   hif_setmodels()

Parameter List

name	type	default	description
vis	string	None	List of input MeasurementSets
reference	variant	None	Reference calibrator field name(s)
refintent	string	AMPLITUDE	Observing intent of reference fields
transfer	variant	None	Transfer calibrator field name(s)
transintent	string	BANDPASS	Observing intent of transfer fields
reffile	string	None	Path to file with fluxes for non-solar system calibrators
normfluxes	bool	True	Normalize lookup fluxes
scalebychan	bool	True	Scale the flux density on a per channel basis or else on a per spw basis
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display commands (True)
acceptresults	bool	True	Automatically accept results into context

Back

hif_tclean

Task Description

Compute clean map
Compute a cleaned image for a particular target source/intent and spectral window.
Keyword arguments:
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets in the context.
default: ''
example: vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
imagename -- Prefix of output images. Defaults to one of the following options depending on the availability of project information.
'{ousstatus uid}.{field}.[{intent}.]s{stage number}.spw{spw}'
'multivis.{field}.[{intent}.]s{stage number}.spw{spw}'
clean boxes and thresholds to use as it goes. For each iteration the output images are:
{prename}.iter{n}.image; cleaned and restored image
{prename}.iter{n}.psf; point spread function (dirty beam)
{prename}.iter{n}.flux; relative sky sensitivity over field
{prename}.iter{n}.flux.pbcoverage; relative pb coverage over field (only for mosaics)
{prename}.iter{n}.model; image of clean components
{prename}.iter{n}.residual; image of residuals
{prename}.iter{n}.cleanmask; image of cleanmask used
default: ''
example: 'test1'

intent -- An intent against which the selected fields are matched. Default means select all data from fields specified by 'field' parameter.
default: ''
example: '', 'TARGET'
field -- Fields id(s) or name(s) to image or mosaic. Must be set.
default:
example: '3C279', 'Centaurus*'
spw -- Spectral window/channels to image. \'\' for all science data.

default: ''
example: '9', '9,11'
spwsel_lsrk -- Spectral window LSRK frequency selection for continuum. {} for all science data.
default: {}
example: {'spw16': '89.1~89.5GHz;90.2~90.3GHz', 'spw18': '101.2~102.1GHz'}
spwsel_topo -- Per MS spectral window TOPO frequency selection for continuum. [] for all science data.
default: {}
example: {['spw16:89.1~89.5GHz;90.2~90.3GHz']}
specmode -- Frequency imaging mode, 'mfs', 'cont', 'cube'. \'\' defaults to
'cube' if intent parameter includes 'TARGET' otherwise 'mfs'.
default: ''
example: 'mfs', 'cont', 'cube'
gridder -- Gridding options, 'standard', 'mosaic'.
Derived as follows:
1. The 'field' parameter is converted into a list of field ids for
each MeasurementSet in 'vis'.
2. If there is more than 1 field id in the list for any MeasurementSet
then gridder is set to 'mosaic', otherwise it will be set
to 'standard'.
default: ''
'standard'
deconvolver -- Minor cycle algorithm e.g. hogbom or clark clean. \'\' defaults to
'hogbom'
outframe -- The reference frame of the output image. The only supported option
is 'LSRK'
default: ''
example: 'LSRK'
imsize -- X and Y image size in pixels). Must be even and contain factors
2,3,5,7 only.
Default derived as follows:
1. Determine 'phasecenter' value and spread of field centers around it.
2. Set size of image to cover spread of field centers plus a border of
width 0.75 * beam radius (to first null).
3. Divide x and y extents by 'cell' values to arrive at the numbers of
pixels required.
default: ''
example: [320,320]
cell -- X and Y cell size. Derived from maximum UV spacing. Details TBD
default ''
example: ['0.5arcsec', '0.5arcsec']
phasecenter -- Direction measure or field id for the mosaic center.
Default derived as follows:
1. Make an array containing all the field centers to be imaged together.
2. Derive the mean direction from the directions array.
default: \'\'
example: 2

stokes --
nchan -- Number of channels or planes in the output image, -1 for all
   default: -1
   example: 128
width -- Width of spectral dimension in frequency, \'\' for default.
   default: \'\'
   example: '7.8125MHz'
nbin -- Channel binning factor
   default: -1
   example: 2
restoringbeam -- Gaussian restoring beam for clean, \'\' for default
   default: \'\'
   example:
hm_masking -- Clean masking mode. Options are 'centralregion',
   'psf', 'psfiter', 'auto', 'manual' and 'none'
   default: 'auto'
   example: 'manual'
hm_sidelobethreshold -- Auto-boxing sidelobetheshold.
hm_noisethreshold -- Auto-boxing noisethreshold.
hm_lownoisethreshold -- Auto-boxing lownoisethreshold.
hm_negativethreshold -- Auto-boxing negativethreshold.
hm_minbeamfrac -- Auto-boxing minbeamfrac.
hm_growiterations -- number of binary dilation iterations for growing the mask.
hm_dogrowprune -- Do pruning on the grow mask
hm_minpercentchange -- Mask size change threshold
mask -- Image mask for hm_masking manual mode. User responsible for
   matching image sizes, coordinates, etc.
   default: \'\'
   example: 'mymask.mask'
niter -- Maximum number of iterations per clean call
   default: 5000
   example: 500
threshold -- Threshold for cleaning
   default: '0.0'
   example: '0.05'
maxncleans -- Maximum number of clean calls
   default: 1
   example: 10
cleancontranges -- Clean continuum frequency ranges in cubes
            default=False
            Options: False, True
calcsb -- Force (re-)calculation of sensitivities and beams
         default=False
         Options: False, True
parallel -- use MPI cluster to clean images
   default: \'automatic\'
--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples:
Make an 'mfs' image of calibrator 3c279 using data in spectral window 1.
The cell size is set to 0.2 arcsec in RA and Dec. Other clean parameters
are derived from heuristics:
hif_tclean(field='3c279', cell='0.2arcsec', spw='1', specmode='mfs')
Make a cube of calibrator 3c279 using data in spectral window 1. The
cube planes will be evenly spaced in frequency in the LSRK frame. Other
clean parameters are derived from heuristics.

hif_tclean(field='3c279', cell='0.2arcsec', spw='1', specmode='cube',
outframe='LSRK')

Parameter List

name	**type**	**default**	**description**
vis	stringArray	None	List of input MeasurementSets, \'\' for default
imagename	string	None	Prefix for image filenames, \'\' for default
intent	string	None	Set of data selection intents, \'\' for all
field	string	None	Set of data selection field names or ids
spw	string	None	Set of data selection spectral window/channels \'\' for all
spwsel_lsrk	any	{}	Dictionary of LSRK spw frequency selections, {} for automatic
spwsel_topo	any	{}	List of TOPO spw frequency selections per MS, [] for automatic
uvrange	any	None	Set of uv ranges, \'\' for all
specmode	string	None	Spectral gridding type (mfs, cont, cube, \'\' for default)
gridder	string	None	Gridding options (standard, mosaic, \'\' for default)
deconvolver	string	None	Minor cycle algorithm (hogbom, clark, mtmfs, \'\' for default)

nterms	int	2	Number of terms when deconvolver=\'mtmfs\'
outframe	string	None	velocity frame of output image (LSRK, \'\' for default)
imsize	intArray	None	X and Y image size in pixels, single value same for both, \'\' for default
cell	stringArray	None	X and Y cell size(s), single value same for both, \'\' for default
phasecenter	any	None	Image center (direction or field index), \'\' for default
stokes	string	I	Stokes Planes to make
nchan	int	-1	Number of channels or planes in output image, -1 = all
start	any	None	Start of output spectral dimension
width	any	None	Width of output spectral channels, \'\' for default
nbin	int	-1	Channel binning factor, -1 for default.
restoringbeam	stringArray	None	Gaussian restoring beam, \'\' for default
hm_masking	string	auto	Pipeline heuristics masking option
hm_sidelobethreshold	double	-999.0	sidelobethreshold * the max sidelobe level
hm_noisethreshold	double	-999.0	noisethreshold * rms in residual image
hm_lownoisethreshold	double	-999.0	lownoisethreshold * rms in residual image
hm_negativethreshold	double	-999.0	negativethreshold * rms in residual image
hm_minbeamfrac	double	-999.0	minimum beam fraction for pruning
hm_growiterations	int	-999	number of binary dilation iterations for growing the mask
hm_dogrowprune	bool	True	Do pruning on the grow mask
hm_minpercentchange	double	-999.0	Mask size change threshold
hm_cleaning	string	rms	Pipeline clean control heuristics
mask	any	None	User mask, \'\' for whole image
niter	int	5000	Maximum number of clean iterations
threshold	double	0.0	Flux level to stop cleaning, must include units: \'1.0mJy\'
tlimit	double	2.0	Times the sensitivity limit for cleaning
masklimit	int	4	Times good mask pixels for cleaning

maxncleans	int	1	Maximum number of clean task calls
cleancontranges	bool	False	Clean continuum frequency ranges in cubes
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
parallel	string	automatic	Clean images using MPI cluster
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hif_transformimagedata

Task Description

Base transformimagedata task
The hif_transformimagedata task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
datacolumn -- Select spectral windows to split. The standard CASA options are
supported
default: 'corrected',
example: 'data', 'model'
replace -- If a split was performed delete the parent MS and remove it from the context.
default: False
example: True or False
clear_pointing -- Clear the pointing table

modify_weights -- Re-initialize the weights

wtmode -- optional weight initialization mode when modify_weights=True
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic transformimagedata task
hif_transformimagedata()

Parameter List

name	type	default	description
vis	stringArray	None	The list of input MeasurementSets
outputvis	string	None	The output MeasurementSet
field	string	None	Set of data selection field names or ids, \'\' for all
intent	string	None	Set of data selection intents, \'\' for all
spw	string	None	Set of data selection spectral window ids \'\' for all
datacolumn	string	corrected	The data columns to process
chanbin	int	1	Channel bin width for spectral averaging
timebin	string	0s	Bin width for time averaging
replace	bool	False	Remove the parent MS and replace with the split MS
clear_pointing	bool	True	Clear the pointing table
modify_weights	bool	False	Re-initialize the weights
wtmode	string	None	Weight initialization mode
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

hif_uvcontfit

Task Description

Fit the continuum in the UV plane
This task estimates the continuum emission by fitting polynomials to
the real and imaginary parts of the spectral windows and channels
selected by spw and exclude spw. This fit represents a model of
the continuum in all channels. Fit orders less than 2 are strongly
recommended. Spw window combination is not currently supported.
Keyword Arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
--- pipeline parameter arguments which can be set in any pipeline mode
solint --
default: 'int'
example: solint='30s'
fitorder -- The fit order of the polynomials.
default: 1
example: fitorder = 0
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets for which the UV continuum fits are
to be generated. Defaults to the list of imaging MeasurementSets specified
in the pipeline import data task.
default '': Compute the continuum fit for all calibration MeasurementSets in the context.
example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
caltable -- The list of output calibration tables one per input MS.
default: '', The output name defaults to the standard pipeline name scheme
example: ['M51.uvcont']
contfile -- The file containing the continuum frequency ranges used for the
continuum fit.
default: '', Defaults first to the file named in the context, next to a file called
'cont.dat' in the pipeline working directory, or ''
example: contfile = 'mycontfile'
field -- The list of field names or field ids for which UV continuum fits are
computed. Defaults to all fields.
default: ''
example: '3C279', '3C279, M82'
intent -- A string containing a comma delimited list of intents against
which the selected fields are matched.

default: '', Defaults to all data with TARGET intent.
example: 'PHASE'
spw -- The list of spectral windows and channels for which uv continuum
fits are computed.
default: '', Defaults to all science spectral windows.
example: '11,13,15,17'
combine -- Data axes to be combined for solving. Axes are 'scan', 'spw', or ''.
This option is currently not supported.
default: '', None.
example: combine= 'scan'

Parameter List

name	type	default	description
vis	stringArray	None	The name of the input visibility file
caltable	stringArray	None	Name of output Mueller matrix calibration table
contfile	string	None	Name of the input file of per source / spw continuum regions
field	string	None	Select field(s) using id(s) or name(s)
intent	string	None	Select intents
spw	string	None	Spectral window / channels for fitting the continuum
combine	string	None	Data axes to combine for the continuum estimation (none, spw and/or scan)
solint	any	int	Time scale for the continuum fit
fitorder	int	1	Polynomial order for the continuum fits
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hif_uvcontsub

Task Description

Subtract the fitted continuum from the data
Apply the UV continuum fit stored in the callibrary to the data.
---- pipeline parameter arguments which can be set in any pipeline mode
applymode -- Calibration apply mode

''='calflagstrict': calibrate data and apply flags from solutions using
the strict flagging convention
'trial': report on flags from solutions, dataset entirely unchanged
'flagonly': apply flags from solutions only, data not calibrated
'calonly': calibrate data only, flags from solutions NOT applied
'calflagstrict':
'flagonlystrict': same as above except flag spws for which calibration is
unavailable in one or more tables (instead of allowing them to pass
uncalibrated and unflagged)
default: '', defaults to 'calonly'
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets in the pipeline context.
default: []
example: ['X227.ms']
field -- A string containing the list of field names or field ids to which
the calibration will be applied. Defaults to all fields in the pipeline
context.
default: ''
example: '3C279', '3C279, M82'
intent -- A string containing the list of intents against which the
selected fields will be matched. Defaults to all supported intents
in the pipeline context.
default: ''
example: '*TARGET*'
spw -- The list of spectral windows and channels to which the calibration
will be applied. Defaults to all science windows in the pipeline
context.
default: ''
example: '17', '11, 15'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned

Description
hif_uvcontsub applies the precomputed uv continuum fit tables stored in the pipeline context to the set of visibility files using predetermined field and spectral window maps and default values for the interpolation schemes.
Users can interact with the pipeline calibration state using the tasks hif_export_calstate and hif_import_calstate.
Issues
Examples
1. Apply the calibration to the target data
hif_uvcontsub()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
applymode	string	None	Calibration mode: ""="calflagstrict","calflag","calflagstrict","trial","flagonly","flagonlystrict", or "calonly"
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

Created at Fri Sep 14 10:04:26 2018 UTC
This page is automatically generated from XML interface definition files.

Summary of ALMA-specific interferometric tasks and parameters

hifa_antpos

Task Description
Derive an antenna position calibration table
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically.

In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'
---- pipeline parameter arguments which can be set in any pipeline mode
hm_antpos -- Heuristics method for retrieving the antenna position corrections. The options are 'online' (not yet implemented), 'manual', and 'file'.
default: 'file'
example: hm_antpos='manual'
antenna -- The list of antennas for which the positions are to be corrected if hm_antpos is 'manual'.
default: none
example: antenna='DV05,DV07'
offsets -- The list of antenna offsets for each antenna in 'antennas'. Each offset is a set of 3 floating point numbers separated by commas, specified in the ITRF frame.
default: none
example: offsets=[0.01, 0.02, 0.03, 0.03, 0.02, 0.01]
antposfile -- The file(s) containing the antenna offsets. Used if hm_antpos is 'file'.
default: 'antennapos.csv'
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- List of input visibility files.
default: []
example: vis=['ngc5921.ms']
caltable -- Name of output gain calibration tables.
default: []
example: caltable=['ngc5921.gcal']
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
The hifa_antpos task corrects the antenna positions recorded in the ASDMs using updated antenna position calibration information determined after the observation was taken.
Corrections can be input by hand, read from a file on disk, or in the future by querying an ALMA database service.
The antenna positions file is in 'csv' format containing 6 comma-delimited

columns as shown below. The default name of this file is 'antennapos.csv'.
Contents of example 'antennapos.csv' file:
ms,antenna,xoffset,yoffset,zoffset,comment
uid___A002_X30a93d_X43e.ms,DV11,0.000,0.010,0.000,"No comment"
uid___A002_X30a93d_X43e.dup.ms,DV11,0.000,-0.010,0.000,"No comment"
The corrections are used to generate a calibration table which is recorded in the pipeline context and applied to the raw visibility data, on the fly to generate other calibration tables, or permanently to generate calibrated visibilities for imaging.
Issues
The hm_antpos 'online' option will be implemented when the observing system provides an antenna position determination service.
Example
1. Correct the position of antenna 5 for all the visibility files in a single pipeline run:
hifa_antpos(antenna='DV05', offsets=[0.01, 0.02, 0.03])
2. Correct the position of antennas for all the visibility files in a single pipeline run using antenna positions files on disk. These files are assumed to conform to a default naming scheme if 'antposfile' is unspecified by the user:
hifa_antpos(hm_antpos='file', antposfile='myantposfile.csv')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltable(s)
hm_antpos	string	file	The antenna position determination method
antenna	string	None	List of antennas to be corrected
offsets	doubleArray	None	List of position corrections one set per antenna
antposfile	string	None	File containing antenna position corrections
pipelinemode	string	automatic	The pipeline operation mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically accept results into context

Back

hifa_bandpassflag

Task Description

Compute bandpass calibration with flagging
Keyword arguments
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'
hm_phaseup -- The pre-bandpass solution phaseup gain heuristics. The options are 'snr' (compute solution required to achieve the specified SNR), 'manual' (use manual solution parameters), and '' (none).
default: 'snr'
example: hm_phaseup='manual'
phaseupsolint -- The phase correction solution interval in CASA syntax. Used when hm_phaseup='manual' or as a default if the hm_phaseup='snr' heuristic computation fails.
default: 'int'
example: phaseupsolint='300s'
phaseupbw -- Bandwidth to be used for phaseup. Defaults to 500MHz. Used when hm_phaseup='manual'.
default: ''
example: phaseupbw='' to use entire bandpass
phaseupbw='500MHz' to use central 500MHz
phaseupsnr -- The required SNR for the phaseup solution. Used only if hm_phaseup='snr'.
default: 20.0
example: phaseupsnr=10.0
phaseupnsols -- The minimum number of phaseup gain solutions. Used only if hm_phaseup='snr'.
default: 2
example: phaseupnsols=4
hm_bandpass -- The bandpass solution heuristics. The options are 'snr' (compute the solution required to achieve the specified SNR), 'smoothed' (simple smoothing heuristics), and 'fixed' (use the user defined parameters for all spws).
solint -- Time and channel solution intervals in CASA syntax.
default: 'inf,7.8125MHz' for hm_bandpass='fixed'
'inf' for hm_bandpass='snr' or 'smoothed'
example: solint='inf,10ch', solint='inf'
maxchannels -- The bandpass solution smoothing factor in channels. The solution interval is bandwidth / 240. Set to 0 for no smoothing.
Used if hm_bandpass='smoothed".

default: 240
example: maxchannels=0
evenbpints -- Force the per spw frequency solint to be evenly divisible
into the spw bandpass if hm_bandpass='snr'.
default: True
example: evenbpints=False
bpsnr -- The required SNR for the bandpass solution. Used only if
hm_bandpass='snr'.
default: 50.0
example: bpsnr=20.0
bpnsols -- The minimum number of bandpass solutions. Used only if
hm_bandpass='snr'.
default: 8
combine -- Data axes to combine for solving. Axes are '', 'scan', 'spw',
'field' or any comma-separated combination.
default: 'scan'
example: combine='scan,field'
minblperant -- Minimum number of baselines required per antenna for each solve
Antennas with fewer baselines are excluded from solutions.
default: 4
minsnr -- Solutions below this SNR are rejected.
default: 3.0
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context.
default: ''
example: vis=['M51.ms']
caltable -- The list of output calibration tables. Defaults to the standard
pipeline naming convention.
default: ''
example: caltable=['M51.bcal']
field -- The list of field names or field ids for which bandpasses are
computed. If undefined (default), it will select all fields.
default: ''
example: field='3C279', field='3C279, M82'
intent -- A string containing a comma delimited list of intents against
which the selected fields are matched. If undefined (default), it
will select all data with the BANDPASS intent.
default: ''
example: intent='*PHASE*'
spw -- The list of spectral windows and channels for which bandpasses are
computed. If undefined (default), it will select all science spectral
windows.
default: ''
example: spw='11,13,15,17'
refant -- Reference antenna names. Defaults to the value(s) stored in the

pipeline context. If undefined in the pipeline context defaults to
the CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'
solnorm -- Normalise the bandpass solutions.
default: False
antnegsig -- Lower sigma threshold for identifying outliers as a result of bad antennas within individual timestamps.
default: 4.0
antpossig -- Upper sigma threshold for identifying outliers as a result of bad antennas within individual timestamps.
default: 4.6
tmantint -- Threshold for maximum fraction of timestamps that are allowed to contain outliers.
default: 0.063
tmint -- Initial threshold for maximum fraction of "outlier timestamps" over "total timestamps" that a baseline may be a part of.
default: 0.085
tmbl -- Initial threshold for maximum fraction of "bad baselines" over "all baselines" that an antenna may be a part of.
default: 0.175
antblnegsig -- Lower sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines, across all timestamps.
default: 3.4
antblpossig -- Upper sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines, across all timestamps.
default: 3.2
relaxed_factor -- Relaxed value to set the threshold scaling factor to under certain conditions (see task description).
default: 2.0
niter -- Maximum number of times to iterate on evaluation of flagging heuristics. If an iteration results in no new flags, then subsequent iterations are skipped.
default: 2
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Examples

1. run with recommended settings to create bandpass solution with flagging
using recommended thresholds:
   hifa_bandpassflag()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltables
intent	string	None	Set of data selection intents
field	string	None	Field ids to use in data selection
spw	string	None	Spectral windows to use in data selection, \'\' for all
antenna	string	None	Set of data selection antenna IDs
hm_phaseup	string	snr	Phaseup before computing the bandpass
phaseupsolint	any	int	Phaseup correction solution interval
phaseupbw	string	None	Bandwidth to use for phaseup
phaseupsnr	double	20.0	SNR for phaseup solution
phaseupnsols	int	2	Minimum number of phaseup gain solutions
hm_bandpass	string	snr	Bandpass solution heuristics
solint	any	inf	Solution intervals
maxchannels	int	240	The smoothing factor in channels
evenbpints	bool	True	Force frequency solint to even bandpass intervals
bpsnr	double	50.0	SNR for bandpass solution
bpnsols	int	8	Minimum number of bandpass solutions
combine	string	scan	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
solnorm	bool	True	Normalise the bandpass solution
antnegsig	double	4.0	Lower sigma threshold for identifying outliers as a result of bad antennas within individual timestamps

antpossig	double	4.6	Upper sigma threshold for identifying outliers as a result of bad antennas within individual timestamps
tmantint	double	0.063	Threshold for maximum fraction of timestamps that are allowed to contain outliers
tmint	double	0.085	Initial threshold for maximum fraction of "outlier timestamps" over "total timestamps" that a baseline may be a part of
tmbl	double	0.175	Initial threshold for maximum fraction of "bad baselines" over "all baselines" that an antenna may be a part of
antblnegsig	double	3.4	Lower sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines (across all timestamps)
antblpossig	double	3.2	Upper sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines (across all timestamps)
relaxed_factor	double	2.0	Relaxed value to set the threshold scaling factor to under certain conditions (see task description)
niter	int	2	Maximum number of times to iterate on evaluation of flagging heuristics.
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

---

hifa_bandpass

Task Description

Compute bandpass calibration solutions
Compute amplitude and phase as a function of frequency for each spectral window in each MeasurementSet.
Previous calibration can be applied on the fly.
Keyword arguments
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

hm_phaseup -- The pre-bandpass solution phaseup gain heuristics. The options
   are 'snr' (compute solution required to achieve the specified SNR),
   'manual' (use manual solution parameters), and '' (none).
   default: 'snr'
   example: hm_phaseup='manual'
phaseupsolint -- The phase correction solution interval in CASA syntax.
   Used when hm_phaseup='manual' or as a default if the hm_phaseup='snr'
   heuristic computation fails.
   default: 'int'
   example: phaseupsolint='300s'
phaseupbw -- Bandwidth to be used for phaseup. Defaults to 500MHz.
   Used when hm_phaseup='manual'.
   default: ''
   example: phaseupbw='' to use entire bandpass
         phaseupbw='500MHz' to use central 500MHz
phaseupsnr -- The required SNR for the phaseup solution. Used only if
   hm_phaseup='snr'.
   default: 20.0
   example: phaseupsnr=10.0
phaseupnsols -- The minimum number of phaseup gain solutions. Used only if
   hm_phaseup='snr'.
   default: 2
   example: phaseupnsols=4
hm_bandpass -- The bandpass solution heuristics. The options are 'snr'
   (compute the solution required to achieve the specified SNR),
   'smoothed' (simple smoothing heuristics), and 'fixed' (use
   the user defined parameters for all spws).
solint -- Time and channel solution intervals in CASA syntax.
   default: 'inf' Used for hm_bandpass='fixed', and as a default
   for the 'snr' and 'smoothed' options.
   default: 'inf,7.8125MHz'
   example: solint='inf,10ch', solint='inf'
maxchannels -- The bandpass solution smoothing factor in channels. The
   solution interval is bandwidth / 240. Set to 0 for no smoothing.
   Used if hm_bandpass='smoothed".
   default: 240
   example: maxchannels=0
evenbpints -- Force the per spw frequency solint to be evenly divisible
   into the spw bandpass if hm_bandpass='snr'.
   default: True
   example: evenbpints=False
bpsnr -- The required SNR for the bandpass solution. Used only if
   hm_bandpass='snr'.
   default: 50.0
   example: bpsnr=20.0
bpnsols -- The minimum number of bandpass solutions. Used only if
   hm_bandpass='snr'.

default: 8
combine -- Data axes to combine for solving. Axes are '', 'scan', 'spw',
'field' or any comma-separated combination.
default: 'scan'
example: combine='scan,field'
minblperant -- Minimum number of baselines required per antenna for each solve.
Antennas with fewer baselines are excluded from solutions.
default: 4
minsnr -- Solutions below this SNR are rejected.
default: 3.0
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context.
default: ''
example: vis=['M51.ms']
caltable -- The list of output calibration tables. Defaults to the standard
pipeline naming convention.
default: ''
example: caltable=['M51.bcal']
field -- The list of field names or field ids for which bandpasses are
computed. Defaults to all fields.
default: ''
example: field='3C279', field='3C279,M82'
intent -- A string containing a comma delimited list of intents against
which the selected fields are matched. Defaults to all data
with bandpass intent.
default: ''
example: intent='*PHASE*'
spw -- The list of spectral windows and channels for which bandpasses are
computed. Defaults to all science spectral windows.
default: ''
example: spw='11,13,15,17'
refant -- Reference antenna names. Defaults to the value(s) stored in the
pipeline context. If undefined in the pipeline context defaults to
the CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'
solnorm -- Normalise the bandpass solutions
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
hifa_bandpass computes a bandpass solution for every specified science
spectral window. By default a 'phaseup' pre-calibration is performed
and applied on the fly to the data, before the bandpass is computed.
The hif_refant task may be used to precompute a prioritized list of
reference antennas.
Issues
There is currently some discussion about whether or not to do an 'ampup'
operations at the same time as the 'phaseup'. This is not required for the
bandpass computation but the amplitude information may provide a useful quality
assessment measure.
The specified minsnr parameter is currently applied to the bandpass
solution computation but not the 'phaseup' computation. Some noisy
solutions in the phaseup may not be properly rejected.
Examples
1. Compute a channel bandpass for all visibility files in the pipeline
context using the CASA reference antenna determination scheme:
hifa_bandpass()
2. Same as the above but precompute a prioritized reference antenna list:
hif_refant()
hifa_bandpass()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna IDs
hm_phaseup	string	snr	Phaseup before computing the bandpass
phaseupsolint	any	int	Phaseup correction solution interval
phaseupbw	string	None	Bandwidth to use for phaseup
phaseupsnr	double	20.0	SNR for phaseup solution
phaseupnsols	int	2	Minimum number of phaseup gain solutions
hm_bandpass	string	snr	Bandpass solution heuristics
solint	any	inf	Solution intervals
maxchannels	int	240	The smoothing factor in channels
evenbpints	bool	True	Force frequency solint to even bandpass intervals
bpsnr	double	50.0	SNR for bandpass solution
bpnsols	int	8	Minimum number of bandpass solutions
combine	string	scan	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	True	Normalise the bandpass solution
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

hifa_bpsolint

Task Description

Compute optimal bandpass calibration solution intervals
Compute the bandpass phaseup time solution interval and bandpass frequency solution interval.
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
phaseupnsr -- The required phaseup gain time interval solution signal to noise.
default: 20.0
example: phaseupsnr=10.0
minphaseupints -- The minimum number of time intervals in the phaseup gain. solution.
default: 2
example: minphaseupints=4
bpnsr -- The required bandpass frequency interval solution signal to noise.
default: 50.0
example: phaseupsnr=20.0
minbpnchan -- The minimum number of frequency intervals in the bandpass solution.
default: 8
example: minbpnchan=16
hm_nantennas -- The heuristics for determines the number of antennas to use in the signal to noise estimate. The options are 'all' and 'unflagged'. The 'unflagged' options is not currently supported.
default: 'all'
example: hm_nantennas='unflagged'
maxfracflagged -- The maximum fraction of an antenna that can be flagged before it is excluded from the signal to noise estimate.
default: 0.90
example: maxfracflagged=0.80
---- pipeline context defined parameter arguments which can be set only in 'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context
default: ''
example: vis=['M82A.ms', 'M82B.ms']
field -- The list of field names of sources to be used for signal to noise estimation. Defaults to all fields with the standard intent.
default: ''

example: field='3C279'
intent -- A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to 'BANDPASS'.
default: 'BANDPASS'
example: intent='PHASE'
spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all the science spectral windows for which there are both 'intent' and TARGET intents.
default: ''
example: spw='13,15'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
The phaseup gain time and bandpass frequency intervals are determined as follows:
o For each data set the list of source(s) to use for bandpass solution signal to noise estimation is compiled based on the values of the field, intent, and spw parameters.
o Source fluxes are determined for each spw and source combination.
o Fluxes in Jy are derived from the pipeline context.
o Pipeline context fluxes are derived from the online flux calibrator catalog, the ASDM, or the user via the flux.csv file.
o If no fluxes are available the task terminates.
o Atmospheric calibration and observations scans are determined for each spw and source combination.
o If intent is set to 'PHASE' are there are no atmospheric scans associated with the 'PHASE' calibrator, 'TARGET' atmospheric scans will be used instead.
o If atmospheric scans cannot be associated with any of the spw and source combinations the task terminates.
o Science spws are mapped to atmospheric spws for each science spw and source combinations.
o If mappings cannot be determined for any of the spws the task terminates
o The median Tsys value for each atmospheric spw and source combination is determined from the SYSCAL table. Medians are computed first by channel, then by antenna, in order to reduce sensitivity to deviant values.
o The science spw parameters, exposure time(s), and integration time(s) are determined.

o The phase up time interval, in time units and number of integrations required to meet the phaseupsnr are computed, along with the phaseup sensitivity in mJy and the signal to noise per integration. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
o Warnings are issued if estimated phaseup gain time solution would contain fewer than minphaseupints solutions
o The frequency interval, in MHz and number of channels required to meet the bpsnr are computed, along with the per channel sensitivity in mJy and the per channel signal to noise. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
o Warnings are issued if estimated bandpass solution would contain fewer than minbpnchan solutions

Examples

1. Estimate the phaseup gain time interval and the bandpass frequency interval required to match the desired signal to noise for bandpass solutions:

```
hifa_bpsolint()
```

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	Set of data selection field names
intent	string	BANDPASS	Set of data selection observing intents
spw	string	None	Set of data selection spectral window ids
phaseupsnr	double	20.0	The required bandpass phaseup signal to noise
minphaseupints	int	2	The minimum number of phaseup intervals in the time solution
evenbpints	bool	False	Force the bandpass frequency solution intervals to be an even number of channels
bpsnr	double	50.0	The required bandpass frequency solution signal to noise
minbpnchan	int	8	The minimum number of channels in the frequency solution
hm_nantennas	string	all	The antenna selection heuristic (unsupported)
maxfracflagged	double	0.90	The maximum fraction of data flagged per antenna (unsupported)
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

hifa_exportdata

Task Description

Prepare interferometry data for export
The hif_exportdata task exports the data defined in the pipeline context
and exports it to the data products directory, converting and or
packing it as necessary.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.

default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in 'interactive mode'
vis -- List of visibility data files for which flagging and calibration information will be exported. Defaults to the list maintained in the pipeline context.
default: []
example: vis=['X227.ms', 'X228.ms']
session -- List of sessions one per visibility file. Currently defaults to a single virtual session containing all the visibility files in vis. In the future, this will default to the set of observing sessions defined in the context.
default: []
example: session=['session1', 'session2']
exportmses -- Export the final MeasurementSets instead of the final flags, calibration tables, and calibration instructions.
default: False
example: exportmses=True
pprfile -- Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'.
default: []
example: pprfile=['PPR_GRB021004.xml']
calintents -- List of calibrator image types to be exported. Defaults to all standard calibrator intents 'BANDPASS', 'PHASE', 'FLUX'.
default: ''
example: calintents='PHASE'
calimages -- List of calibrator images to be exported. Defaults to all calibrator images recorded in the pipeline context.
default: []
example: calimages=['3C454.3.bandpass', '3C279.phase']
targetimages -- List of science target images to be exported. Defaults to all science target images recorded in the pipeline context.
default: []
example: targetimages=['NGC3256.band3', 'NGC3256.band6']
products_dir -- Name of the data products subdirectory.
default: './'
example: products_dir='../products'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Description
The hif_exportdata task exports the data defined in the pipeline context and exports it to the data products directory, converting and or packing it as necessary.
The current version of the task exports the following products
o an XML file containing the pipeline processing request
o a tar file per ASDM / MS containing the final flags version
o a text file per ASDM / MS containing the final calibration apply list
o a FITS image for each selected calibrator source image
o a FITS image for each selected science target source image
o a tar file per session containing the caltables for that session
o a tar file containing the file web log
o a text file containing the final list of CASA commands
Issues
Support for merging the calibration state information into the pipeline context / results structure and retrieving it still needs to be added.
Support for merging the clean results into the pipeline context / results structure and retrieving it still needs to be added.
Support for creating the final pipeline results entity still needs to be added.

Session information is not currently handled by the pipeline context. By default all ASDMs are combined into one session.
Examples
1. Export the pipeline results for a single session to the data products directory:

```
!mkdir ../products
hif_exportdata(products_dir='../products')
```

2. Export the pipeline results to the data products directory specify that only the gain calibrator images be saved:

```
!mkdir ../products
hif_exportdata(products_dir='../products', calintents='*PHASE*')
```

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
imaging_products_only	bool	False	Export science target imaging products only
exportmses	bool	False	Export final MeasurementSets instead of flags and caltables
pprfile	string	None	The pipeline processing request file to be exported
calintents	string	None	The calibrator source target intents to be exported
calimages	stringArray	None	List of calibrator images to be exported
targetimages	stringArray	None	List of target images to be exported
products_dir	string	None	The data products directory
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifa_flagdata

Task Description

Do basic flagging
The hifa_flagdata data performs basic flagging operations on a list of
MeasurementSets.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
autocorr -- Flag autocorrelation data.
default: True
shadow -- Flag shadowed antennas.
default: True
scan -- Flag a list of specified scans.
default: True

scannumber -- A string containing a comma delimited list of scans to be
   flagged.
   default: ''
   example: scannumber='3,5,6'
intents -- A string containing a comma delimited list of intents against
   which the scans to be flagged are matched.
   default: 'POINTING,FOCUS,ATMOSPHERE,SIDEBAND'
   example: intents='*BANDPASS*'
edgespw -- Flag the edge spectral window channels.
   default: True
fracspw -- Fraction of the baseline correlator TDM edge channels to be flagged.
   default: 0.03125
fracspwfps -- Fraction of the ACA correlator TDM edge channels to be flagged.
   default: 0.048387
online -- Apply the online flags.
   default: True
fileonline -- File containing the online flags. These are computed by the
   h_init or hif_importdata data tasks. If the online flags files
   are undefined a name of the form 'msname.flagonline.txt' is assumed.
   default: ''
template -- Apply flagging templates
   default: True
filetemplate -- The name of a text file that contains the flagging template
   for RFI, birdies, telluric lines, etc. If the template flags files
   is undefined a name of the form 'msname.flagtemplate.txt' is assumed.
   default: ''
hm_tbuff -- The heuristic for computing the default time interval padding
parameter. The options are 'halfint' and 'manual'. In 'halfint' mode tbuff
is set to half the maximum of the median integration time of the science
and calibrator target observations. The value of 0.048 seconds is
subtracted from the lower time limit to accommodate the behavior of the
ALMA Control system.
   default: 'halfint'
tbuff -- The time in seconds used to pad flagging command time intervals if
   hm_tbuff='manual'. The default in manual mode is no flagging.
   default: [0.0,0.0]
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets defined in the pipeline context.
   default: ''
flagbackup -- Back up any pre-existing flags.
   default: True
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not
execute (False).
   default: True

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hifa_flagdata data performs basic flagging operations on a list of measurements including:
o applying online flags
o applying a flagging template
o autocorrelation data flagging
o shadowed antenna data flagging
o scan-based flagging by intent or scan number
o edge channel flagging
Issues
Examples
1. Do basic flagging on a MeasurementSet:

hifa_flagdata()
2. Do basic flagging on a MeasurementSet flagging additional scans selected
by number as well:
hifa_flagdata(scannumber='13,18')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets to flag
autocorr	bool	True	Flag autocorrelation data
shadow	bool	True	Flag shadowed antennas
scan	bool	True	Flag specified scans
scannumber	string	None	List of scans to be flagged
intents	string	POINTING,FOCUS,ATMOSPHERE,SIDEBAND	List of intents of scans to be flagged
edgespw	bool	True	Flag edge channels

fracspw	double	0.03125	Fraction of baseline correlator edge channels to be flagged
fracspwfps	double	0.048387	Fraction of ACA correlator edge channels to be flagged
online	bool	True	Apply the online flags
fileonline	string	None	File of online flags to be applied
template	bool	True	Apply a flagging template
filetemplate	stringArray	None	File that contains the flagging template
hm_tbuff	string	halfint	The time buffer computation heuristic
tbuff	any	[0.0,0.0]	List of time buffers (sec) to pad timerange in flag commands
qa0	bool	True	QA0 flags
qa2	bool	True	QA2 flags
pipelinemode	string	automatic	The pipeline operating mode
flagbackup	bool	True	Backup pre-existing flags before applying new ones
dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hifa_flagtargets

Task Description

Do science target flagging
The hifa_flagtargets task performs basic flagging operations on a list of
science target MeasurementSets.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
template -- Apply flagging templates.
default: True
filetemplate -- The name of a text file that contains the flagging template
for issues with the science target data etc. If the template flags files
is undefined a name of the form 'msname_flagtargetstemplate.txt' is assumed.
default: ''
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets defined in the pipeline context.
default: ''
flagbackup -- Back up any pre-existing flags.
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not
execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hifa_flagdata data performs basic flagging operations on a list of
measurements including:
o applying a flagging template
Issues
Examples
1. Do basic flagging on a science target MeasurementSet:

```
hifa_flagtargets()
```

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets to flag
template	bool	True	Apply a flagging template
filetemplate	stringArray	None	File that contains the flagging template
pipelinemode	string	automatic	The pipeline operating mode
flagbackup	bool	False	Backup pre-existing flags before applying new ones
dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifa_fluxcalflag

Task Description

Locate and flag line regions in solar system flux calibrators
Fills the model column with the model visibilities.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
threshold -- If the fraction of an spw occupied by line regions is greater
then threshold flag the entire spectral window.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets defined in the pipeline context.
default: []
field -- The list of field names or field ids for which the models are
to be set. Defaults to all fields with intent 'AMPLITUDE'.
default: ''
example: field='3C279', field='3C279, M82'
intent -- A string containing a comma delimited list of intents against
which the selected fields are matched. Defaults to all data
with amplitude intent.
default: ''

example: intent='AMPLITUDE'
spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.
default: ''
example: spw='11,13,15,17'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
Search the built-in solar system flux calibrator line catalog for overlaps with the science spectral windows. Generate a list of line overlap regions and flagging commands.
Issues
Examples
1. Locate known lines in any solar system object flux calibrators:
hifa_fluxcalflag()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	List of field names or ids
intent	string	None	Observing intent of flux calibrators
spw	string	None	List of spectral window ids
pipelinemode	string	automatic	The pipeline operating mode
threshold	double	0.75	Threshold for flagging the entire spw
appendlines	bool	False	Append user defined line regions to the line dictionary
linesfile	string	None	File containing user defined lines
applyflags	bool	True	Apply the computed flag commands
dryrun	bool	False	Run the task (False) or display the commands (True)
acceptresults	bool	True	Automatically accept results into the context

hifa_fluxdb

Task Description

Connect to flux calibrator database
Connect to the ALMA flux calibrator database.
Keyword arguments
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
vis -- List of input visibility files.
default: none
example: vis=['ngc5921.ms']
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
Connect to the ALMA flux calibrator database
Issues
Example
1. Connect to the ALMA flux calibrator database:
hifa_fluxdb()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

hifa_gaincalsnr

Task Description

Compute gaincal signal to noise ratios per spw
Compute the per scan gaincal solution signal to noise ratio per science spw.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In interactive mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'
phasesnr -- The required gaincal solution signal to noise.
default: 25.0
example: phasesnr=20.0
bwedgefrac -- The fraction of the bandwidth edges that is flagged.
default: 0.03125
example: bwedgefrac=0.0
hm_nantennas -- The heuristics for determines the number of antennas to use
in the signal to noise estimate. The options are 'all' and 'unflagged'.
The 'unflagged' options is not currently supported.
default: 'all'
example: hm_nantennas='unflagged'
maxfracflagged -- The maximum fraction of an antenna that can be flagged
before it is excluded from the signal to noise estimate.
default: 0.90
example: maxfracflagged=0.80
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context.
default: ''
example: vis=['M82A.ms', 'M82B.ms']
field -- The list of field names of sources to be used for signal to noise
estimation. Defaults to all fields with the standard intent.
default: ''
example: field='3C279'
intent -- A string containing a comma delimited list of intents against which
the selected fields are matched. Defaults to 'PHASE'.
default: 'PHASE'
example: intent='BANDPASS'
spw -- The list of spectral windows and channels for which gain solutions are

computed. Defaults to all the science spectral windows for which there are both 'intent' and TARGET intents.
default: ''
example: spw='13,15'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Description
The gaincal solution signal to noise is determined as follows:
o For each data set the list of source(s) to use for the per scan gaincal solution signal to noise estimation is compiled based on the values of the field, intent, and spw parameters.
o Source fluxes are determined for each spw and source combination.
  o Fluxes in Jy are derived from the pipeline context.
  o Pipeline context fluxes are derived from the online flux calibrator catalog, the ASDM, or the user via the flux.csv file.
  o If no fluxes are available the task terminates.
o Atmospheric calibration and observations scans are determined for each spw and source combination.
  o If intent is set to 'PHASE' are there are no atmospheric scans associated with the 'PHASE' calibrator, 'TARGET' atmospheric scans will be used instead.
  o If atmospheric scans cannot be associated with any of the spw and source combinations the task terminates.
o Science spws are mapped to atmospheric spws for each science spw and source combinations.
  o If mappings cannot be determined for any of the spws the task terminates.
o The median Tsys value for each atmospheric spw and source combination is determined from the SYSCAL table. Medians are computed first by channel, then by antenna, in order to reduce sensitivity to deviant values.
o The science spw parameters, exposure time(s), and integration time(s) are determined.
o The per scan sensitivity and signal to noise estimates are computed per science spectral window. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
o The QA score is based on how many signal to noise estimates greater than the requested signal to noise ration can be computed.
Examples
1. Estimate the per scan gaincal solution sensitivities and signal to noise

ratios for all the science spectral windows:
hifa_gaincalsnr()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	Set of data selection field names
intent	string	PHASE	Set of data selection observing intents
spw	string	None	Set of data selection spectral window ids
phasesnr	double	25.0	The signal to noise minimum
bwedgefrac	double	0.03125	The fraction of the bandwidth edge that is flagged
hm_nantennas	string	all	The antenna selection heuristic (unsupported)
maxfracflagged	double	0.90	The maximum fraction of data flagged per antenna (unsupported)
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

hifa_gfluxscaleflag

Task Description

Derive the flux density scale with flagging
Keyword arguments
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context.
default: ''

example: vis=['M51.ms']
phaseupsolint -- The phase correction solution interval in CASA syntax.
default: 'int'
example: phaseupsolint='300s'
solint -- Time and channel solution intervals in CASA syntax.
default: 'inf'
example: solint='inf,10ch', solint='inf'
minsnr -- Solutions below this SNR are rejected.
default: 2.0
refant -- Reference antenna names. Defaults to the value(s) stored in the pipeline context. If undefined in the pipeline context defaults to the CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'
antnegsig -- Lower sigma threshold for identifying outliers as a result of bad antennas within individual timestamps.
default: 4.0
antpossig -- Upper sigma threshold for identifying outliers as a result of bad antennas within individual timestamps.
default: 4.6
tmantint -- Threshold for maximum fraction of timestamps that are allowed to contain outliers.
default: 0.063
tmint -- Initial threshold for maximum fraction of "outlier timestamps" over "total timestamps" that a baseline may be a part of.
default: 0.085
tmbl -- Initial threshold for maximum fraction of "bad baselines" over "all baselines" that an antenna may be a part of.
default: 0.175
antblnegsig -- Lower sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines, across all timestamps.
default: 3.4
antblpossig -- Upper sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines, across all timestamps.
default: 3.2
relaxed_factor -- Relaxed value to set the threshold scaling factor to under certain conditions (see task description).
default: 2.0
niter -- Maximum number of times to iterate on evaluation of flagging heuristics. If an iteration results in no new flags, then subsequent iterations are skipped.
default: 2
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).

default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. run with recommended settings to create flux scale calibration with flagging
using recommended thresholds:
hifa_gfluxscaleflag()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
phaseupsolint	any	int	Phaseup correction solution interval
solint	any	inf	Solution intervals
minsnr	double	2.0	Reject solutions below this SNR
refant	string	None	Reference antenna names
antnegsig	double	4.0	Lower sigma threshold for identifying outliers as a result of bad antennas within individual timestamps
antpossig	double	4.6	Upper sigma threshold for identifying outliers as a result of bad antennas within individual timestamps
tmantint	double	0.063	Threshold for maximum fraction of timestamps that are allowed to contain outliers
tmint	double	0.085	Initial threshold for maximum fraction of "outlier timestamps" over "total timestamps" that a baseline may be a part of
tmbl	double	0.175	Initial threshold for maximum fraction of "bad baselines" over "all baselines" that an antenna may be a part of
antblnegsig	double	3.4	Lower sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines (across all timestamps)
antblpossig	double	3.2	Upper sigma threshold for identifying outliers as a result of "bad baselines" and/or "bad antennas" within baselines (across all timestamps)
relaxed_factor	double	2.0	Relaxed value to set the threshold scaling factor to under certain conditions (see task description)
niter	int	2	Maximum number of times to iterate on evaluation of flagging heuristics.
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

hifa_gfluxscale

Task Description

Derive flux density scales from standard calibrators
Derive flux densities for point source transfer calibrators using flux models
for reference calibrators.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the users can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
phaseupsolint -- Time solution intervals in CASA syntax for the phase solution.
default: 'int'
example: phaseupsolint='inf', phaseupsolint='int', phaseupsolint='100sec'
solint -- Time solution intervals in CASA syntax for the amplitude solution.
default: 'inf'
example: solint='inf', solint='int', solint='100sec'
minsnr -- Minimum signal to noise ratio for gain calibration solutions.
default: 2.0
example: minsnr=1.5, minsnr=0.0
hm_resolvedcals - Heuristics method for handling resolved calibrators. The
options are 'automatic' and 'manual'. In automatic mode antennas closer
to the reference antenna than the uv distance where visibilities fall to
'peak_fraction' of the peak are used. In manual mode the antennas specified
in 'antenna' are used.
antenna -- A comma delimited string specifying the antenna names or ids to be
used for the fluxscale determination. Used in hm_resolvedcals='manual' mode.
default: ''
example: antenna='DV16,DV07,DA12,DA08'
peak_fraction -- The limiting UV distance from the reference antenna for antennas
to be included in the flux calibration. Defined as the point where the
calibrator visibilities have fallen to 'peak_fraction' of the peak value.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
specified in the pipeline context
default: ''
example: ['M32A.ms', 'M32B.ms']
reference -- A string containing a comma delimited list of field names
defining the reference calibrators. Defaults to field names with
intent '*AMP*'.
default: ''
example: reference='M82,3C273'

transfer -- A string containing a comma delimited list of field names
defining the transfer calibrators. Defaults to field names with
intent '*PHASE*'.
default: ''
example: transfer='J1328+041,J1206+30'
refintent -- A string containing a comma delimited list of intents
used to select the reference calibrators. Defaults to *AMP*.
default: ''
example: refintent='', refintent='*AMP*'
refspwmap -- Vector of spectral window ids enabling scaling across
spectral windows. Defaults to no scaling.
default: []
example: refspwmap=[1,1,3,3] - (4 spws, reference fields in 1 and 3, transfer
fields in 0,1,2,3
reffile -- Path to a file containing flux densities for calibrators unknown to
CASA. Values given in this file take precedence over the CASA-derived
values for all calibrators except solar system calibrators. By default the
path is set to the CSV file created by hifa_importdata, consisting of
catalogue fluxes extracted from the ASDM and / or edited by the user.
default: ''
example: reffile='', reffile='working/flux.csv'
transintent -- A string containing a comma delimited list of intents
defining the transfer calibrators. Defaults to *PHASE*.
default: ''
example: transintent='', transintent='*PHASE*'
refant -- A string specifying the reference antenna(s). By default this is read
from the context.
default: ''
example: refant='DV05'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not
execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
Derive flux densities for point source transfer calibrators using flux models
for reference calibrators.
Flux values are determined by:
o computing complex gain phase only solutions for all the science spectral
windows using the calibrator data selected by the 'reference' and
'refintent' parameters and the 'transfer' and 'transintent' parameters,
and the value of the 'phaseupsolint' parameter.

o computing complex amplitude only solutions for all the science spectral windows using calibrator data selected with 'reference' and 'refintent' parameters and the 'transfer' and 'transintent' parameters, the value of the 'solint' parameter.
o transferring the flux scale from the reference calibrators to the transfer calibrators using refspwmap for windows without data in the reference calibrators
o extracted the computed flux values from the CASA logs and inserting them into the MODEL_DATA column.

Resolved calibrators are handled via antenna selection either automatically, hm_resolvedcals='automatic' or manually, hm_resolvedcals='manual'. In the former case antennas closer to the reference antenna than the uv distance where visibilities fall to 'peak_fraction' of the peak are used. In manual mode the antennas specified in 'antenna' are used.

Note that the flux corrected calibration table computed internally is not currently used in later pipeline apply calibration steps.

Issues

Should we add a spw window selection option here?

The code which extracts the flux scales from the logs needs to be replaced with code which uses the values returned from the CASA fluxscale task.

Examples

1. Compute flux values for the phase calibrator using model data from the amplitude calibrator:

```
hifa_gfluxscale()
```

Parameter List

name	type	default	description
vis	string	None	List of input MeasurementSets
reference	variant	None	Reference calibrator field name(s)
transfer	variant	None	Transfer calibrator field name(s)
refintent	string	None	Observing intent of reference fields
transintent	string	None	Observing intent of transfer fields
refspwmap	intArray	None	Map across spectral window boundaries
reffile	string	None	Path to file with fluxes for non-solar system calibrators
phaseupsolint	any	int	Phaseup correction solution interval
solint	any	inf	Amplitude correction solution interval
minsnr	double	2.0	Minimum SNR for gain solutions
refant	string	None	The name or ID of the reference antenna
hm_resolvedcals	string	automatic	The resolved calibrators heuristics method
antenna	string	None	Antennas to be used in fluxscale
peak_fraction	double	0.2	Fraction of peak visibility at uv-distance limit of antennas to be used
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display commands (True)
acceptresults	bool	True	Automatically accept results into context

Back

---

hifa_imageprecheck

Task Description

Base imageprecheck task
The hifa_imageprecheck task.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSes, or tar files of MSes. If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']

calcsb -- Force (re-)calculation of sensitivities and beams.
default: False
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic imageprecheck task:
hifa_imageprecheck()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
calcsb	bool	False	Force (re-)calculation of sensitivities
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifa_importdata

Task Description

Imports data into the interferometry pipeline
The hifa_importdata task loads the specified visibility data into the pipeline
context unpacking and / or converting it as necessary.
Keyword arguments

---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
    MSes, or tar files of MSes. If ASDM files are specified, they will be
    converted to MS format.
    default: []
    example: vis=['X227.ms', 'asdms.tar.gz']
session -- List of sessions to which the visibility files belong. Defaults
    to a single session containing all the visibility files, otherwise
    a session must be assigned to each vis file.
    default: []
    example: session=['session_1', 'session_2']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
    determines the values of all context defined pipeline inputs automatically.
    In 'interactive' mode the user can set the pipeline context defined
    parameters manually. In 'getinputs' mode the user can check the settings of
    all pipeline parameters without running the task.
    default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
asis -- ASDM tables to convert as is.
    default: 'SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR'
    example: asis='Receiver', asis=''
process_caldevice -- Ingest the ASDM caldevice table.
    default: False
    example: process_caldevice=True
overwrite -- Overwrite existing MSes on output.
    default: False
nocopy -- When importing an MS, disable copying of the MS to the working
    directory.
    default: False
bdfflags -- Apply BDF flags on line.
    default: True
ocorr_mode -- Read in cross- and auto-correlation data(ca), cross-
    correlation data only (co), or autocorrelation data only (ao).
    default: ca
lazy -- Use the lazy filter import.
    default: False
dbservice -- Use online flux catalog on import.
    default: True
ocorr_mode -- Read in cross- and auto-correlation data(ca), cross-
    correlation data only (co), or autocorrelation data only (ao).
    default: ca
createmms -- Create a multi-MeasurementSet ('true') ready for full parallel
    processing, or a standard MeasurementSet ('false'). The default setting
    ('automatic') creates an MMS if running in a cluster environment.
    default: 'false'
--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Load an ASDM list in the ../rawdata subdirectory into the context:
hifa_importdata(vis=['../rawdata/uid___A002_X30a93d_X43e',
'../rawdata/uid_A002_x30a93d_X44e'])
2. Load an MS in the current directory into the context:
hifa_importdata(vis=[uid___A002_X30a93d_X43e.ms])
3. Load a tarred ASDM in ../rawdata into the context:
hifa_importdata(vis=['../rawdata/uid___A002_X30a93d_X43e.tar.gz'])
4. Check the hif_importdata inputs, then import the data:
myvislist = ['uid___A002_X30a93d_X43e.ms', 'uid_A002_x30a93d_X44e.ms']
hifa_importdata(vis=myvislist, pipelinemode='getinputs')
hifa_importdata(vis=myvislist)
5. Load an ASDM but check the results before accepting them into the context.
results = hifa_importdata(vis=['uid___A002_X30a93d_X43e.ms'],
acceptresults=False)
results.accept()
6. Run in dryrun mode before running for real:
results = hifa_importdata(vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)
results = hifa_importdata(vis=['uid___A002_X30a93d_X43e.ms'])

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of visibility data sessions
pipelinemode	string	automatic	The pipeline operating mode
asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR	Extra ASDM tables to convert as is
process_caldevice	bool	False	Import the caldevice table from the ASDM
overwrite	bool	False	Overwrite existing files on import
nocopy	bool	False	Disable copying of MS to working directory
bdfflags	bool	True	Apply BDF flags on import
asimaging	bool	False	Import MeasurementSets as imaging MeasurementSets
lazy	bool	False	Use the lazy filler import
dbservice	bool	True	Use the online flux catalog
ocorr_mode	string	ca	ALMA default set to ca
createmms	string	false	Create an MMS
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifa_linpolcal

Task Description

Compute polarization calibration
Compute a polarization calibration.
Keyword arguments
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline

determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets in the context.
CURRENTLY THE LIST MUST CONTAIN 1 MEASUREMENT SET.
default: ''
example: vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets, \'\' for default
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection intents
g0table	string	None	Name of table holding G0 gain - not accounting for source pol
delaytable	string	None	Name of table holding cross-hand delay
xyf0table	string	None	Name of table holding residual X-Y phase spectrum and source Q and U
g1table	string	None	Name of table holding G1 gain - accounting for source pol
df0table	string	None	Name of table holding instrument polarization gain
refant	string	None	Reference antenna names
spw	string	None	Set of data selection spectral window/channels
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hifa_restoredata

Task Description

Restore flagged and calibration interferometry data from a pipeline run
The hifa_restoredata task restores flagged and calibrated MeasurementSets
from archived ASDMs and pipeline flagging and calibration date products.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of raw visibility data files to be restored. Assumed to be
in the directory specified by rawdata_dir.
default: None

example: vis=['uid___A002_X30a93d_X43e']
session -- List of sessions one per visibility file.
default: []
example: session=['session_3']
products_dir -- Name of the data products directory. Currently not used.
default: '../products'
example: products_dir='myproductspath'
rawdata_dir -- Name of the rawdata subdirectory.
default: '../rawdata'
example: rawdata_dir='myrawdatapath'
lazy -- Use the lazy filler option.
default: False
example: lazy=True
bdfflags -- Set the BDF flags.
default: True
example: bdfflags=False
ocorr_mode -- Set ocorr_mode.
default: 'ca'
example: ocorr_mode='ca'
asis -- Set list of tables to import as is.
default: 'SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR'
example: asis='Source Receiver'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hifa_restoredata restores flagged and calibrated data from archived
ASDMs and pipeline flagging and calibration data products. Pending archive
retrieval support hifa_restoredata assumes that the required products
are available in the rawdata_dir in the format produced by the
hifa_exportdata task.
hifa_restoredata assumes that the following entities are available in the raw
data directory.
o the ASDMs to be restored
o for each ASDM in the input list
o a compressed tar file of the final flagversions file, e.g.
uid___A002_X30a93d_X43e.ms.flagversions.tar.gz
o a text file containing the applycal instructions, e.g.
uid___A002_X30a93d_X43e.ms.calapply.txt
o a compressed tar file containing the caltables for the parent session,

e.g. uid___A001_X74_X29.session_3.caltables.tar.gz
hifa_restoredata performs the following operations:
o imports the ASDM(s))
o removes the default MS.flagversions directory created by the filler
o restores the final MS.flagversions directory stored by the pipeline
o restores the final set of pipeline flags to the MS
o restores the final calibration state of the MS
o restores the final calibration tables for each MS
o applies the calibration tables to each MS
Issues
Examples
1. Restore the pipeline results for a single ASDM in a single session:
hifa_restoredata(vis=['uid___A002_X30a93d_X43e'], session=['session_1'],
ocorr_mode='ca')

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
products_dir	string	../products	The archived pipeline data products directory
copytoraw	bool	True	Copy calibration and flagging tables to raw data directory
rawdata_dir	string	../rawdata	The rawdata directory
lazy	bool	False	Use the lazy filler option
bdfflags	bool	True	Set the BDF flags
ocorr_mode	string	ca	Correlation import mode
asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR	List of tables to import asis
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

hifa_session_bandpass

Task Description

Compute bandpass calibration solutions
Compute amplitude and phase as a function of frequency for each spectral window in each MeasurementSet.
Previous calibration can be applied on the fly.
Keyword arguments
--- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
hm_phaseup -- The pre-bandpass solution phaseup gain heuristics. The options are 'snr' (compute solution required to achieve the specified SNR), 'manual' (use manual solution parameters), and '' (none).
default: 'snr'
example: hm_phaseup='manual'
phaseupsolint -- The phase correction solution interval in CASA syntax. Used when hm_phaseup='manual' or as a default if the hm_phaseup='snr' heuristic computation fails.
default: 'int'
example: phaseupsolint='300s'
phaseupbw -- Bandwidth to be used for phaseup. Defaults to 500MHz. Used when hm_phaseup='manual'.
default: ''
example: phaseupbw='' to use entire bandpass, phaseupbw='500MHz' to use central 500MHz
phaseupsnr -- The required SNR for the phaseup solution. Used only if hm_phaseup='snr'.
default: 20.0
example: phaseupsnr=10.0
phaseupnsols -- The minimum number of phaseup gain solutions. Used only if hm_phaseup='snr'.
default: 2
example: phaseupnsols=4
hm_bandpass -- The bandpass solution heuristics. The options are 'snr' (compute the solution required to achieve the specified SNR), 'smoothed' (simple smoothing heuristics), and 'fixed' (use the user defined parameters for all spws).
solint -- Time and channel solution intervals in CASA syntax.
default: 'inf' Used for hm_bandpass='fixed', and as a default

for the ’snr’ and ’smoothed’ options.
default: ’inf,7.8125MHz’
example: solint=’inf,10ch’, solint=’inf’
maxchannels -- The bandpass solution smoothing factor in channels. The
solution interval is bandwidth / 240. Set to 0 for no smoothing.
Used if hm_bandpass=’smoothed".
default: 240
example: 0
evenbpints -- Force the per spw frequency solint to be evenly divisible
into the spw bandpass if hm_bandpass=’snr’.
default: True
example: evenbpints=False
bpsnr -- The required SNR for the bandpass solution. Used only if
hm_bandpass=’snr’
default: 50.0
example: bpsnr=20.0
bpnsols -- The minimum number of bandpass solutions. Used only if
hm_bandpass=’snr’.
default: 8
hm_bandtype -- The type of bandpass. The options are ’channel’ and
’polynomial’ for CASA bandpass types = ’B’ and ’BPOLY’ respectively.
combine -- Data axes to combine for solving. Axes are ’’, ’scan’, ’spw’,
’field’ or any comma-separated combination.
default: ’scan’
example: combine=’scan,field’
minblperant -- Minimum number of baselines required per antenna for each solve
Antennas with fewer baselines are excluded from solutions. Used for
hm_bandtype=’channel’ only.
default: 4
minsnr -- Solutions below this SNR are rejected. Used for hm_bandtype=
’channel’ only.
default: 3.0
---- pipeline context defined parameter arguments which can be set only in
’interactive mode’
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context.
default: ’’
example: vis=[’M51.ms’]
caltable -- The list of output calibration tables. Defaults to the standard
pipeline naming convention.
default: ’’
example: caltable=[’M51.bcal’]
field -- The list of field names or field ids for which bandpasses are
computed. Defaults to all fields.
default: ’’
example: field=’3C279’, field=’3C279, M82’
intent -- A string containing a comma delimited list of intents against

which the selected fields are matched. Defaults to all data
with bandpass intent.
default: ''
example: intent='*PHASE*'
spw -- The list of spectral windows and channels for which bandpasses are
computed. Defaults to all science spectral windows.
default: ''
example: spw='11,13,15,17'
refant -- Reference antenna names. Defaults to the value(s) stored in the
pipeline context. If undefined in the pipeline context defaults to
the CASA reference antenna naming scheme.
default: ''
example: refant='DV01', refant='DV06,DV07'
solnorm -- Normalise the bandpass solutions.
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
hifa_session_bandpass computes a bandpass solution for every specified science
spectral window. By default a 'phaseup' pre-calibration is performed
and applied on the fly to the data, before the bandpass is computed.
The hif_refant task may be used to precompute a prioritized list of
reference antennas.
Issues
There is currently some discussion about whether or not to do an 'ampup'
operations at the same time as the 'phaseup'. This is not required for the
bandpass computation but the amplitude information may provide a useful quality
assessment measure.
The specified minsnr parameter is currently applied to the bandpass
solution computation but not the 'phaseup' computation. Some noisy
solutions in the phaseup may not be properly rejected.
Examples
1. Compute a channel bandpass for all visibility files in the pipeline
context using the CASA reference antenna determination scheme:
hifa_session_bandpass()
2. Same as the above but precompute a prioritized reference antenna list:
hif_refant()
hifa_session_bandpass()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna IDs
hm_phaseup	string	snr	Phaseup before computing the bandpass
phaseupsolint	any	int	Phaseup correction solution interval
phaseupbw	string	None	Bandwidth to use for phaseup
phaseupsnr	double	20.0	SNR for phaseup solution
phaseupnsols	int	2	Minimum number of phaseup gain solutions
hm_bandpass	string	snr	Bandpass solution heuristics
solint	any	inf	Solution intervals
maxchannels	int	240	The smoothing factor in channels
evenbpints	bool	True	Force frequency solint to even bandpass intervals
bpsnr	double	50.0	SNR for bandpass solution
bpnsols	int	8	Minimum number of bandpass solutions
hm_bandtype	string	channel	Bandpass solution type
combine	string	scan	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	True	Normalise the bandpass solution
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
degamp	variant	None	Degree for polynomial amplitude solution
degphase	variant	None	Degree for polynomial phase solution
pipelinemode	string	automatic	The pipeline operating mode

dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context
parallel	string	automatic	Execute using CASA HPC functionality, if available.

Back

---

hifa_spwphaseup

Task Description

Compute phase calibration spw map and per spw phase offsets
Compute the gain solutions.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
hm_spwmapmode -- The spectral window mapping mode. The options are: 'auto',
'combine', 'simple', and 'default'. In 'auto' mode hifa_spwphaseup
estimates the SNR of the phase calibrator observations and uses these
estimates to choose between 'combine' mode (low SNR) and 'default' mode
(high SNR). In combine mode all spectral windows are combined and mapped to
one spectral window. In 'simple' mode narrow spectral windows are mapped to
wider ones using an algorithm defined by 'maxnarrowbw', 'minfracmaxbw', and
'samebb'. In 'default' mode the spectral window map defaults to the
standard one to one mapping.
default: 'auto'
example: hm_spwmapmode='combine'
maxnarrowbw -- The maximum bandwidth defining narrow spectral windows. Values
must be in CASA compatible frequency units.
default: '300MHz'
example: maxnarrowbw=''
minfracmaxbw -- The minimum fraction of the maximum bandwidth in the set of
spws to use for matching.
default: 0.8
example: minfracmaxbw=0.75
samebb -- Match within the same baseband if possible.
default: True
example: samebb=False
phasesnr -- The required gaincal solution signal to noise.
default: 32.0
example: phaseupsnr=20.0

bwedgefrac -- The fraction of the bandwidth edges that is flagged.
    default: 0.03125
    example: bwedgefrac=0.0
hm_nantennas -- The heuristics for determines the number of antennas to use
    in the signal to noise estimate. The options are 'all' and 'unflagged'.
    The 'unflagged' options is not currently supported.
    default: 'all'
    example: hm_nantennas='unflagged'
maxfracflagged -- The maximum fraction of an antenna that can be flagged
    before it is excluded from the signal to noise estimate.
    default: 0.90
    example: maxfracflagged=0.80
combine -- Data axes to combine for solving. Options are '', 'scan', 'spw',
    'field' or any comma-separated combination.
    default: ''
    example: combine=''
minblperant -- Minimum number of baselines required per antenna for each solve.
    Antennas with fewer baselines are excluded from solutions.
    default: 4
    example: minblperant=2
minsnr -- Solutions below this SNR are rejected.
    default: 3.0
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
    MeasurementSets specified in the pipeline context.
    default: ''
    example: vis=['M82A.ms', 'M82B.ms']
caltable -- The list of output calibration tables. Defaults to the standard
    pipeline naming convention.
    default: ''
    example: caltable=['M82.gcal', 'M82B.gcal']
field -- The list of field names or field ids for which phase offset solutions
    are to be computed. Defaults to all fields with the default intent.
    default: ''
    example: field='3C279', field='3C279, M82'
intent -- A string containing a comma delimited list of intents against
    which the selected fields are matched. Defaults to the BANDPASS
    observations.
    default: ''
    example: intent='PHASE'
spw -- The list of spectral windows and channels for which gain solutions are
    computed. Defaults to all the science spectral windows.
    default: ''
    example: spw='13,15'
refant -- Reference antenna name(s) in priority order. Defaults to most recent
    values set in the pipeline context. If no reference antenna is defined in

the pipeline context the CASA defaults are used.
default: ''
example: refant='DV01', refant='DV05,DV07'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
hifa_spwphaseup performs two functions:
- o determines the spectral window mapping mode for the phase vs time calibrations and computes spectral window map that will be used to apply those calibrations
- o computes the per spectral window phase offset table that will be applied to the data to remove mean phase differences between the spectral windows

If hm_spwmapmode = 'auto' the spectral window map is computed using the following algorithm:
o estimate the per spectral window per scan signal to noise ratio of the phase calibrator observations
o if the signal to noise of any single phase calibration spectral window is less than the value of 'phasesnr' hm_spwmapmode defaults to 'combine'
o if all phase calibrator spectral windows meet the low signal to noise criterion then hm_spwmapmode defaults to default'
o if the phase calibrator signal to noise values cannot be computed for any reason, for example there is no flux information, then hm_spwmapmode defaults to 'combine'
If hm_spwmapmode = 'combine' hifa_spwphaseup maps all the science windows to a single science spectral window. For example if the list of science spectral windows is [9, 11, 13, 15] then all the science spectral windows in the data will be combined and mapped to the science window 9 in the combined phase vs time calibration table.
If hm_spwmapmode = 'simple', a mapping from narrow science to wider science spectral windows is computed using the following algorithm:
o construct a list of the bandwidths of all the science spectral windows
o determine the maximum bandwidth in this list maxbandwidth
o for each science spectral window with bandwidth less than maxbandwidth
- o construct a list of spectral windows with bandwidths greater than minfracmaxbw * maxbandwidth
- o select the spectral window in this list whose band center most closely matches the band center of the narrow spectral window
- o preferentially match within the same baseband if samebb is True

If hm_spwmapmode = 'default' the spw mapping is assumed to be one to one.
Phase offsets per spectral window are determined by computing a phase only gain calibration on the selected data, normally the high signal to noise bandpass calibrator observations, using the solution interval 'inf'.
At the end of the task the spectral window map and the phase offset calibration table in the pipeline are stored in the context for use by later tasks.
Examples
1. Compute the default spectral window map and the per spectral window phase offsets:
hifa_spwphaseup()
2. Compute the default spectral window map and the per spectral window phase offsets set the spectral window mapping mode to 'simple':
hifa_spwphaseup(hm_spwmapmode='simple')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltables
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
hm_spwmapmode	string	auto	The spw mapping mode
maxnarrowbw	string	300MHz	The maximum bandwidth defining narrow spectral windows
minfracmaxbw	double	0.8	The minimum fraction of the maximum bandpass for spw matching
samebb	bool	True	Match within the same baseband if possible
phasesnr	double	32.0	The minimum snr for triggering spw combination in auto spw mapping mode
bwedgefrac	double	0.03125	The fraction of the bandwidth edge that is flagged
hm_nantennas	string	all	The antenna selection heuristic
maxfracflagged	double	0.90	The maximum fraction of data flagged per antenna
combine	string	None	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
minblperant	int	4	Minimum baselines per antenna required for solve
minsnr	double	3.0	Reject solutions below this SNR
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

hifa_timegaincal

Task Description

Determine temporal gains from calibrator observations
Compute the gain solutions.
Keyword arguments
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings of
all pipeline parameters without running the task.
default: 'automatic'
calsolint -- Time solution interval in CASA syntax for calibrator source
solutions.
default: 'int'
example: calsolint='inf', calsolint='int', calsolint='100sec'
targetsolint -- Time solution interval in CASA syntax for target source
solutions.
default: 'inf'
example: targetsolint='inf', targetsolint='int', targetsolint='100sec'
combine -- Data axes to combine for solving. Options are '', 'scan', 'spw',
'field' or any comma-separated combination.
default: ''
example: combine=''
minblperant -- Minimum number of baselines required per antenna for each solve.
Antennas with fewer baselines are excluded from solutions.
default: 4
example: minblperant=2
calminsnr -- Solutions below this SNR are rejected for calibrator solutions.
default: 2.0
targetminsnr -- Solutions below this SNR are rejected for science target
solutions.
default: 3.0
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of
MeasurementSets specified in the pipeline context.
default: ''
example: vis=['M82A.ms', 'M82B.ms']
calamptable -- The list of output diagnostic calibration amplitude tables for
the calibration targets. Defaults to the standard pipeline naming
convention.
default: ''
example: calamptable=['M82.gacal', 'M82B.gacal']
offsetstable -- The list of output diagnostic phase offset tables for the

calibration targets. Defaults to the standard pipeline naming convention.
default: ''
example: offsetstable=['M82.offsets.gacal', 'M82B.offsets.gacal']
calphasetable -- The list of output calibration phase tables for the
calibration targets. Defaults to the standard pipeline naming convention.
default: ''
example: calphasetable=['M82.gcal', 'M82B.gcal']
amptable -- The list of output calibration amplitude tables for the
calibration and science targets.
Defaults to the standard pipeline naming convention.
default: ''
example: amptable=['M82.gcal', 'M82B.gcal']
targetphasetable -- The list of output phase calibration tables for the science
targets. Defaults to the standard pipeline naming convention.
default: ''
example: targetphasetable=['M82.gcal', 'M82B.gcal']
field -- The list of field names or field ids for which gain solutions are to
be computed. Defaults to all fields with the standard intent.
default: ''
example: field='3C279', field='3C279, M82'
intent -- A string containing a comma delimited list of intents against which
the selected fields are matched. Defaults to the equivalent of
'AMPLITUDE,PHASE,BANDPASS'.
default: ''
example: intent='', intent='PHASE'
spw -- The list of spectral windows and channels for which gain solutions are
computed. Defaults to all science spectral windows.
default: ''
example: spw='3C279', spw='3C279, M82'
smodel -- Point source Stokes parameters for source model (experimental)
Defaults to using standard MODEL_DATA column data.
default: []
example: smodel=[1,0,0,0] - (I=1, unpolarized)
refant -- Reference antenna name(s) in priority order. Defaults to most recent
values set in the pipeline context. If no reference antenna is defined in
the pipeline context use the CASA defaults.
default: ''
example: refant='DV01', refant='DV05,DV07'
solnorm -- Normalise the gain solutions.
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not
execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True

Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The complex gains are derived from the data column (raw data) divided by the model column (usually set with hif_setjy). The gains are obtained for the specified solution intervals, spw combination and field combination. One gain solution is computed for the science targets and one for the calibrator targets.
Good candidate reference antennas can be determined using the hif_refant task.
Previous calibrations that have been stored in the pipeline context are applied on the fly. Users can interact with these calibrations via the hif_export_calstate and hif_import_calstate tasks.
Issues
Examples
1. Compute standard per scan gain solutions that will be used to calibrate the target:
hifa_timegaincal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
calamptable	stringArray	None	List of diagnostic output amplitude caltables for calibrator targets
offsetstable	stringArray	None	List of diagnostic output phase offset caltables for calibrator targets
calphasetable	stringArray	None	List of output phase caltables for calibrator targets
targetphasetable	stringArray	None	List of output phase caltables for science targets
amptable	stringArray	None	List of output amp caltables for science targets
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
calsolint	any	int	Phase solution interval for calibrator sources
targetsolint	any	inf	Phase solution interval for science target sources
combine	string	None	Data axes which to combine for solve (scan, spw, and/or field)
refant	string	None	Reference antenna names
solnorm	bool	False	Normalize average solution amplitudes to 1.0
minblperant	int	4	Minimum baselines per antenna required for solve
calminsnr	double	2.0	Reject solutions below this SNR for calibrator solutions
targetminsnr	double	3.0	Reject solutions below this SNR for science solutions
smodel	doubleArray	None	Point source Stokes parameters for source model
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

hifa_tsysflag

Task Description

Flag deviant system temperature measurements
Flag deviant system temperatures for ALMA interferometry measurements.
Flag all deviant system temperature measurements in the system temperature calibration table by running a sequence of flagging tests, each designed to look for a different type of error.
If a file with manual Tsys flags is provided with the 'filetemplate' parameter, then these flags are applied prior to the evaluation of the flagging heuristics listed below.
The tests are:
1. Flag Tsys spectra with high median values
2. Flag Tsys spectra with high median derivatives. This is meant to spot spectra that are 'ringing'.
3. Flag the edge channels of the Tsys spectra in each SpW.
4. Flag Tsys spectra whose shape is different from that associated with the BANDPASS intent.
5. Flag 'birdies'.
6. Flag the Tsys spectra of all antennas in a timestamp and spw if proportion of antennas already flagged in this timestamp and spw exceeds a threshold, and flag Tsys spectra for all antennas and all timestamps in a spw, if proportion of antennas that are already entirely flagged in all timestamps exceeds a threshold.
Keyword arguments
--- Pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
flag_nmedian -- True to flag Tsys spectra with high median value.
default: True
fnm_limit -- Flag spectra with median value higher than fnm_limit * median of this measure over all spectra.
default: 2.0
fnm_byfield -- Evaluate the nmedian metric separately for each field.
default: True

flag_derivative -- True to flag Tsys spectra with high median derivative.
default: True
fd_max_limit -- Flag spectra with median derivative higher than fd_max_limit * median of this measure over all spectra.
default: 5.0
flag_edgechans -- True to flag edges of Tsys spectra.

default: True
fe_edge_limit -- Flag channels whose channel to channel difference > fe_edge_limit * median across spectrum.
default: 3.0
flag_fieldshape -- True to flag Tsys spectra with a radically different shape to those of the ff_refintent.
default: True
ff_refintent -- Data intent that provides the reference shape for 'flag_fieldshape'.
default: 'BANDPASS'
ff_max_limit -- Flag Tsys spectra with 'fieldshape' metric values > ff_max_limit.
default: 5.0
flag_birdies -- True to flag channels covering sharp spectral features.
default: True
fb_sharps_limit -- Flag channels bracketing a channel to channel difference > fb_sharps_limit.
default: 0.05
flag_toomany -- True to flag Tsys spectra for which a proportion of antennas for given timestamp and/or proportion of antennas that are entirely flagged in all timestamps exceeds their respective thresholds.
default: True
tmf1_limit -- Flag Tsys spectra for all antennas in a timestamp and spw if proportion of antennas already flagged in this timestamp and spw exceeds tmf1_limit.
default: 0.666
tmef1_limit -- Flag Tsys spectra for all antennas and all timestamps in a spw, if proportion of antennas that are already entirely flagged in all timestamps exceeds tmef1_limit.
default: 0.666
metric_order -- Order in which to evaluate the flagging metrics that are enables. Disabled metrics are skipped.
default: 'nmedian,derivative,edgechans,fieldshape,birdies,toomany'
normalize_tsys -- True to create a normalized Tsys table that is used to evaluate the Tsys flagging metrics. All newly found flags are also applied to the original Tsys caltable that continues to be used for subsequent calibration.
default: False
filetemplate -- The name of a text file that contains the manual Tsys flagging template. If the template flags file is undefined, a name of the form 'msname.flagtsystemplate.txt' is assumed.
default: ''
--- Pipeline context defined parameter arguments which can be set only in 'interactive mode'
caltable -- List of input Tsys calibration tables.
default: [] - Use the table currently stored in the pipeline context.
example: caltable=['X132.ms.tsys.s2.tbl']

--- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Flag Tsys measurements using currently recommended tests:
hifa_tsysflag()
2. Flag Tsys measurements using all recommended tests apart from that
using the 'fieldshape' metric:

hifa_tsysflag(flag_fieldshape=False)

Parameter List

name	**type**	**default**	**description**
vis	stringArray	None	List of input MeasurementSets (Not used)
caltable	stringArray	None	List of input caltables
flag_nmedian	bool	True	True to flag Tsys spectra with high median value
fnm_limit	double	2.0	Flag spectra with median greater than fnm_limit * median over all spectra
fnm_byfield	bool	True	Evaluate the nmedian metric separately for each field.
flag_derivative	bool	True	True to flag Tsys spectra with high median derivative

fd_max_limit	double	5.0	Flag spectra with median derivative higher than fd_max_limit * median of this measure over all spectra
flag_edgechans	bool	True	True to flag edges of Tsys spectra
fe_edge_limit	double	3.0	Flag channels whose channel to channel difference greater than fe_edge_limit * median across spectrum
flag_fieldshape	bool	True	True to flag Tsys spectra with a radically different shape to those of the ff_refintent
ff_refintent	string	BANDPASS	Data intent providing the reference shape for \'flag_fieldshape\'
ff_max_limit	double	5.0	Flag Tsys spectra with \'fieldshape\' metric greater than ff_max_limit
flag_birdies	bool	True	True to flag channels covering sharp spectral features
fb_sharps_limit	double	0.05	Flag channels bracketing a channel to channel difference greater than fb_sharps_limit

flag_toomany	bool	True	True to flag Tsys spectra for which a proportion of timestamps or proportion of antennas that are entirely flagged exceeds their respective thresholds.
tmf1_limit	double	0.666	Flag all Tsys spectra within a timestamp for an antenna if proportion flagged already exceeds tmf1_limit
tmef1_limit	double	0.666	Flag all Tsys spectra for all antennas in a spw, if proportion of antennas that are already entirely flagged in all timestamps exceeds tmef1_limit
metric_order	string	nmedian,derivative,edgechans,fieldshape,birdies,toomany	Order in which to evaluate the flagging metric(s); inactive metrics are skipped.
normalize_tsys	bool	False	Normalize Tsys prior to computing the flagging metric(s)
filetemplate	stringArray	None	File containing manual Tsys flags to apply.
pipelinemode	string	automatic	The pipeline operations mode

dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

---

hifa_wvrgcalflag

Task Description

This task will first identify for each vis whether it includes at least 3 antennas with Water Vapor Radiometer (WVR) data, and that the fraction of WVR antennas / all antennas exceeds the minimum threshold (ants_with_wvr_thresh).
If there are not enough WVR antennas by number and/or fraction, then no WVR caltable is created and no WVR calibration will be applied to the corresponding vis. If there are enough WVR antennas, then the task proceeds as follows for each valid vis:
First, generate a gain table based on the Water Vapor Radiometer data for each vis.
Second, apply the WVR calibration to the data specified by 'flag_intent', calculate flagging 'views' showing the ratio 'phase-rms with WVR / phase-rms without WVR' for each scan. A ratio < 1 implies that the phase noise is improved, a score > 1 implies that it is made worse.
Third, search the flagging views for antennas with anomalous high values. If any are found then recalculate the WVR calibration with the 'wvrflag' parameter set to ignore their data and interpolate results from other antennas according to 'maxdistm' and 'minnumants'.
Fourth, after flagging, if the remaining unflagged antennas with WVR number fewer than 3, or represent a smaller fraction of antennas than the minimum threshold (ants_with_wvr_thresh), then the WVR calibration file is rejected and will not be merged into the context, i.e. not be used in subsequent calibration.
Fifth, if the overall QA score for the final WVR correction of a vis file is greater than the value in 'accept_threshold' then make available the wvr calibration file for merging into the context and use in the subsequent reduction.
Keyword arguments
vis -- List of input visibility files.
default: none, in which case the vis files to be used will be read from the context
example: vis=['ngc5921.ms']

caltable -- List of output gain calibration tables.
    default: none, in which case the names of the caltables will be
        generated automatically
    example: caltable='ngc5921.wvr'
offsetstable -- List of input temperature offsets table files to subtract from
    WVR measurements before calculating phase corrections.
    default: none, in which case no offsets are applied
    example: offsetstable=['ngc5921.cloud_offsets']
hm_toffset -- If 'manual', set the 'toffset' parameter to the user-specified
    value. If 'automatic', set the 'toffset' parameter according to the
    date of the MeasurementSet; toffset=-1 if before 2013-01-21T00:00:00
    toffset=0 otherwise.
    default: 'automatic'
toffset -- Time offset (sec) between interferometric and WVR data.
    default: 0
segsource -- If True calculate new atmospheric phase correction
    coefficients for each source, subject to the constraints of
    the 'tie' parameter. 'segsource' is forced to be True if
    the 'tie' parameter is set to a non-empty value by the
    user or by the automatic heuristic.
    default: True
hm_tie -- If 'manual', set the 'tie' parameter to the user-specified value.
    If 'automatic', set the 'tie' parameter to include with the
    target all calibrators that are within 15 degrees of it:
    if no calibrators are that close then 'tie' is left empty.
    default: 'automatic'
tie -- Use the same atmospheric phase correction coefficients when
    calculating the WVR correction for all sources in the 'tie'. If 'tie'
    is not empty then 'segsource' is forced to be True. Ignored unless
    hm_tie='manual'.
    default: []
    example: tie=['3C273,NGC253', 'IC433,3C279']
sourceflag -- Flag the WVR data for these source(s) as bad and do not produce
    corrections for it. Requires segsource=True.
    default: []
    example: sourceflag=['3C273']
nsol -- Number of solutions for phase correction coefficients during this
    observation, evenly distributed in time throughout the observation. It
    is used only if segsource=False because if segsource=True then the
    coefficients are recomputed whenever the telescope moves to a new
    source (within the limits imposed by 'tie').
    default: 1
disperse -- Apply correction for dispersion.
    default: False
wvrflag -- Flag the WVR data for these antenna(s) as bad and replace its data
    with interpolated values.
    default: []

example: wvrflag=['DV03','DA05','PM02']
hm_smooth -- If 'manual' set the 'smooth' parameter to the user-specified value.
If 'automatic', run the wvrgcal task with the range of 'smooth' parameters
required to match the integration time of the WVR data to that of the
interferometric data in each spectral window.
smooth -- Smooth WVR data on this timescale before calculating the correction.
Ignored unless hm_smooth='manual'.
default: ''
scale -- Scale the entire phase correction by this factor.
default: 1
maxdistm -- Maximum distance in meters of an antenna used for interpolation
from a flagged antenna.
default: -1 (automatically set to 100m if >50% of antennas are 7m
antennas without WVR and otherwise set to 500m)
example: maxdistm=550
minnumants -- Minimum number of nearby antennas (up to 3) used for
interpolation from a flagged antenna.
default: 2
example: minnumants=3
mingoodfrac -- Minimum fraction of good data per antenna.
default: 0.8
example: mingoodfrac=0.7
refant -- Ranked comma delimited list of reference antennas.
default: ''
example: refant='DV02,DV06'
flag_intent -- The data intent(s) on whose WVR correction results the search
for bad WVR antennas is to be based.
A 'flagging view' will be calculated for each specified intent, in each
spectral window in each vis file.
Each 'flagging view' will consist of a 2-d image with dimensions
['ANTENNA', 'TIME'], showing the phase noise after the WVR
correction has been applied.
If flag_intent is left blank, the default, the flagging views will be
derived from data with the default bandpass calibration intent i.e.
the first in the list BANDPASS, PHASE, AMPLITUDE for which the
MeasurementSet has data.
default: ''
qa_intent -- The list of data intents on which the WVR correction is to be
tried as a means of estimating its effectiveness.
A QA 'view' will be calculated for each specified intent, in each spectral
window in each vis file.
Each QA 'view' will consist of a pair of 2-d images with dimensions
['ANTENNA', 'TIME'], one showing the data phase-noise before the
WVR application, the second showing the phase noise after (both 'before'
and 'after' images have a bandpass calibration applied as well).
An overall QA score is calculated for each vis file, by dividing the
'before' images by the 'after' and taking the median of the result. An

overall score of 1 would correspond to no change in the phase noise,
a score > 1 implies an improvement.
If the overall score for a vis file is less than the value in
'accept_threshold' then the WVR calibration file is not made available for
merging into the context for use in the subsequent reduction.
default: 'BANDPASS,PHASE'

qa_bandpass_intent -- The data intent to use for the bandpass calibration in
the qa calculation. The default is blank to allow the underlying bandpass
task to select a sensible intent if the dataset lacks BANDPASS data.
default: ''

accept_threshold -- The phase-rms improvement ratio
(rms without WVR / rms with WVR) above which the wrvg file will be
accepted into the context for subsequent application.
default: 1.0

ants_with_wvr_thresh -- this threshold sets the minimum fraction of antennas
that should have WVR data for WVR calibration and flagging to proceed; the
same threshold is used to determine, after flagging, whether there remain
enough unflagged antennas with WVR data for the WVR calibration to be
applied.
default: 0.2
example: ants_with_wvr_thresh=0.5

Example

1. Compute the WVR calibration for all the MeasurementSets:
hifa_wvrgcalflag(hm_tie='automatic')

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility files
caltable	stringArray	None	List of output gain calibration tables
offsetstable	stringArray	None	List of input temperature offsets table files
hm_toffset	string	automatic	Toffset computation heuristic method
toffset	double	0	Time offset (sec) between IF and WVR data
segsource	bool	True	Compute new coefficient calculation for each source
sourceflag	stringArray	None	Flag the WVR data for these source(s)
hm_tie	string	automatic	Tie computation heuristics method
tie	stringArray	None	Sources for which to use the same atmospheric phase correction coefficients

nsol	int	1	Number of solutions for phase correction coefficients
disperse	bool	False	Apply correction for dispersion
wvrflag	stringArray	None	Flag the WVR data for these antenna(s) replace with interpolated values
hm_smooth	string	automatic	Smoothing computation heuristics method
smooth	string	None	Smooth WVR data on the given timescale before calculating the correction
scale	double	1.	Scale the entire phase correction by this factor
maxdistm	double	-1	Maximum distance (m) of an antenna used for interpolation for a flagged antenna
minnumants	int	2	Minimum number of near antennas (up to 3) required for interpolation
mingoodfrac	double	0.8	Minimum fraction of good data per antenna
refant	string	None	Ranked list of reference antennas
flag_intent	string	None	Data intents to use in detecting and flagging bad WVR antennas
qa_intent	string	BANDPASS,PHASE	Data intents to use in estimating the effectiveness of the WVR correction
qa_bandpass_intent	string	None	Data intent to use for the bandpass calibration in the qa calculation
accept_threshold	double	1.0	Improvement ratio (phase-rms without WVR / phase-rms with WVR) above which wvrg calibration file will be accepted
flag_hi	bool	True	True to flag high figure of merit outliers
fhi_limit	double	10.0	Flag figure of merit values higher than limit * MAD
fhi_minsample	int	5	Minimum number of samples for valid MAD estimate
ants_with_wvr_thresh	double	0.2	Minimum fraction of unflagged antennas that need to have WVR for calibration to proceed.
pipelinemode	string	automatic	The pipeline operating mode

dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Back

---

hifa_wvrgcal

Task Description

Generate a gain table based on the Water Vapor Radiometer data in each vis file. By applying the wvr calibration to the data specified by 'qa_intent' and 'qa_spw', calculate a QA score to indicate its effect on interferometric data; a score > 1 implies that the phase noise is improved, a score < 1 implies that it is made worse. If the score is less than 'accept_threshold' then the wvr gain table is not accepted into the context for subsequent use.

Keyword arguments

vis -- List of input visibility files.
default: none, in which case the vis files to be used will be read from the context.
example: vis=['ngc5921.ms']

caltable -- List of output gain calibration tables.
default: none, in which case the names of the caltables will be generated automatically.
example: caltable='ngc5921.wvr'

offsetstable -- List of input temperature offsets table files to subtract from WVR measurements before calculating phase corrections.
default: none, in which case no offsets are applied.
example: offsetstable=['ngc5921.cloud_offsets']

hm_toffset -- If 'manual', set the 'toffset' parameter to the user-specified value.
If 'automatic', set the 'toffset' parameter according to the date of the MeasurementSet; toffset=-1 if before 2013-01-21T00:00:00 toffset=0 otherwise.
default: 'automatic'

toffset -- Time offset (sec) between interferometric and WVR data.
default: 0

segsource -- If True calculate new atmospheric phase correction coefficients for each source, subject to the constraints of the 'tie' parameter.
'segsource' is forced to be True if the 'tie' parameter is set to a non-empty value by the user or by the automatic heuristic.
default: True

hm_tie -- If 'manual', set the 'tie' parameter to the user-specified value.
If 'automatic', set the 'tie' parameter to include with the target all calibrators that are within 15 degrees of it:
if no calibrators are that close then 'tie' is left empty.

default: 'automatic'
tie -- Use the same atmospheric phase correction coefficients when
calculating the wvr correction for all sources in the 'tie'. If 'tie'
is not empty then 'segsource' is forced to be True. Ignored unless
hm_tie='manual'.
default: []
example: ['3C273,NGC253', 'IC433,3C279']
sourceflag -- Flag the WVR data for these source(s) as bad and do not produce
corrections for it. Requires segsource=True.
default: []
example: ['3C273']
nsol -- Number of solutions for phase correction coefficients during this
observation, evenly distributed in time throughout the observation. It
is used only if segsource=False because if segsource=True then the
coefficients are recomputed whenever the telescope moves to a new source
(within the limits imposed by 'tie').
default: 1
disperse -- Apply correction for dispersion.
default: False
wvrflag -- Flag the WVR data for the listed antennas as bad and replace
their data with values interpolated from the 3 nearest antennas with
unflagged data.
default: []
example: ['DV03','DA05','PM02']
hm_smooth -- If 'manual' set the 'smooth' parameter to the user-specified value.
If 'automatic', run the wvrgcal task with the range of 'smooth' parameters
required to match the integration time of the wvr data to that of the
interferometric data in each spectral window.
smooth -- Smooth WVR data on this timescale before calculating the correction.
Ignored unless hm_smooth='manual'.
default: ''
scale -- Scale the entire phase correction by this factor.
default: 1
maxdistm -- Maximum distance in meters of an antenna used for interpolation
from a flagged antenna.
default: -1 (automatically set to 100m if >50% of antennas are 7m
antennas without WVR and otherwise set to 500m)
example: maxdistm=550
minnumants -- Minimum number of nearby antennas (up to 3) used for
interpolation from a flagged antenna.
default: 2
example: minnumants=3
mingoodfrac -- Minimum fraction of good data per antenna.
default: 0.8
refant -- Ranked comma delimited list of reference antennas.
default: ''
example: refant='DV01,DV02'

qa_intent -- The list of data intents on which the wvr correction is to be tried as a means of estimating its effectiveness.
A QA 'view' will be calculated for each specified intent, in each spectral window in each vis file.
Each QA 'view' will consist of a pair of 2-d images with dimensions ['ANTENNA', 'TIME'], one showing the data phase-noise before the wvr application, the second showing the phase noise after (both 'before' and 'after' images have a bandpass calibration applied as well).
An overall QA score is calculated for each vis file, by dividing the 'before' images by the 'after' and taking the median of the result. An overall score of 1 would correspond to no change in the phase noise, a score > 1 implies an improvement.
If the overall score for a vis file is less than the value in 'accept_threshold' then the wvr calibration file is not made available for merging into the context for use in the subsequent reduction.
If you do not want any QA calculations then set qa_intent=''.
default: ''
example: qa_intent='PHASE'

qa_bandpass_intent -- The data intent to use for the bandpass calibration in the qa calculation. The default is blank to allow the underlying bandpass task to select a sensible intent if the dataset lacks BANDPASS data.
default: ''

qa_spw -- The SpW(s) to use for the qa calculation, in the order that they should be tried. Input as a comma-separated list. The default is blank, in which case the task will try SpWs in order of decreasing median sky opacity.
default: ''

accept_threshold -- The phase-rms improvement ratio (rms without wvr / rms with wvr) above which the wrvg file will be accepted into the context for subsequent application.
default: 1.0

Example

1. Compute the WVR calibration for all the MeasurementSets:
hifa_wvrgcal(hm_tie='automatic')

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility files
caltable	stringArray	None	List of output gain calibration tables
offsetstable	stringArray	None	List of input temperature offsets table files
hm_toffset	string	automatic	Toffset computation heuristic method
toffset	double	0	Time offset (sec) between IF and WVR data

segsource	bool	True	Compute new coefficient calculation for each source
sourceflag	stringArray	None	Flag the WVR data for these source(s)
hm_tie	string	automatic	Tie computation heuristics method
tie	stringArray	None	Sources for which to use the same atmospheric phase correction coefficients
nsol	int	1	Number of solutions for phase correction coefficients
disperse	bool	False	Apply correction for dispersion
wvrflag	stringArray	None	Flag the WVR data for these antenna(s) replace with interpolated values
hm_smooth	string	automatic	Smoothing computation heuristics method
smooth	string	None	Smooth WVR data on the given timescale before calculating the correction
scale	double	1.	Scale the entire phase correction by this factor
maxdistm	double	-1	Maximum distance (m) of an antenna used for interpolation for a flagged antenna.
minnumants	int	2	Minimum number of near antennas (up to 3) required for interpolation
mingoodfrac	double	0.8	Minimum fraction of good data per antenna
refant	string	None	Ranked list of reference antennas
qa_intent	string	None	Data intents to use in estimating the effectiveness of the wvr correction
qa_bandpass_intent	string	None	Data intent to use for the bandpass calibration in the qa calculation
qa_spw	string	None	Data SpW(s) to use in estimating the effectiveness of the wvr correction
accept_threshold	double	1.0	Improvement ratio (phase-rms without wvr / phase-rms with wvr) above which wvrg calibration file will be accepted
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

Created at Fri Sep 14 10:04:26 2018 UTC
This page is automatically generated from XML interface definition files.

Summary of VLA-specific interferometric tasks and parameters

hifv_applycals

Task Description

Applycals
The hifv_applycals task
Apply precomputed calibrations to the data.
---- pipeline parameter arguments which can be set in any pipeline mode
applymode -- Calibration apply mode
''='calflagstrict': calibrate data and apply flags from solutions using
the strict flagging convention
'trial': report on flags from solutions, dataset entirely unchanged
'flagonly': apply flags from solutions only, data not calibrated
'calonly': calibrate data only, flags from solutions NOT applied
'calflagstrict':
'flagonlystrict': same as above except flag spws for which calibration is
unavailable in one or more tables (instead of allowing them to pass
uncalibrated and unflagged)
default: ''
flagsum -- Compute before and after flagging statistics summaries.
default: True
flagdetailedsum -- Compute detailed before and after flagging statistics summaries
if flagsum is True.
default: True
gainmap -- Mode to map gainfields to a particular list of scans
default: False
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
in the pipeline context.
default: []

example: ['X227.ms']
field -- A string containing the list of field names or field ids to which
the calibration will be applied. Defaults to all fields in the pipeline
context.
default: ''
example: '3C279', '3C279, M82'
intent -- A string containing the list of intents against which the
selected fields will be matched. Defaults to all supported intents
in the pipeline context.
default: ''
example: '*TARGET*'
spw -- The list of spectral windows and channels to which the calibration
will be applied. Defaults to all science windows in the pipeline
context.
default: ''
example: '17', '11, 15'
antenna -- The list of antennas to which the calibration will be applied.
Defaults to all antennas. Not currently supported.
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned
Description
hifv_applycals applies the precomputed calibration tables stored in the pipeline
context to the set of visibility files using predetermined field and
spectral window maps and default values for the interpolation schemes.
Users can interact with the pipeline calibration state using the tasks
hif_export_calstate and hif_import_calstate.
Issues
There is some discussion about the appropriate values of calwt. Given
properly scaled data, the correct value should be the CASA default of True.
However at the current time ALMA is suggesting that calwt be set to True for
applying observatory calibrations, e.g. antenna positions, WVR, and system
temperature corrections, and to False for applying instrument calibrations,
e.g. bandpass, gain, and flux.
Examples
1. Run the final applycals stage of the VLA CASA pipeline.
hifv_applycals()

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
applymode	string	None	Calibration mode: ""="calflagstrict","calflag","calflagstrict","trial","flagonly","flagonlystrict", or "calonly"
flagbackup	bool	True	Backup the flags before the apply
flagsum	bool	True	Compute before and after flagging summary statistics
flagdetailedsum	bool	True	Compute detailed flagging statistics
gainmap	bool	False	Mode to map gainfields to scans.
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

hifv_checkflag

Task Description

Run flagdata in rflag mode
The hifv_checkflag task runs flagdata in rflag mode
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
checkflagmode -- blank string default use of rflag on bandpass and delay calibrators
-- use string 'semi' after hifv_semiFinalBPdcals() for executing rflag on calibrators
-- use string 'bpd', for the bandpass and delay calibrators:
execute rflag on all calibrated cross-hand corrected data;
extend flags to all correlations
execute rflag on all calibrated parallel-hand residual data;

extend flags to all correlations
execute tfcrop on all calibrated cross-hand corrected data,
per visibility; extend flags to all correlations
execute tfcrop on all calibrated parallel-hand corrected data,
per visibility; extend flags to all correlations
-- use string 'allcals', for all the other calibrators, with delays and BPcal applied:
similar procedure as 'bpd' mode, but uses corrected data throughout
-- use string 'target', for the target data:
similar procedure as 'allcals' mode, but with a higher SNR cutoff
for rflag to avoid flagging data due to source structure, and
with an additional series of tfcrop executions to make up for
the higher SNR cutoff in rflag
-- VLASS specific modes include 'bpd-vlass', 'allcals-vlass', and 'target-vlass'
which calculate thresholds to user per spw/field/scan (action='calculate', then,
per baseband/field/scan, replace all spw thresholds above the median with the median,
before re-running rflag with the new thresholds. This has the effect of
lowering the thresholds for spws with RFI to be closer to the RFI-free
thresholds, and catches more of the RFI.
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Run RFLAG with associated heuristics in the VLA CASA pipeline.
hifv_checkflag()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
checkflagmode	string	None	String name of the mode to run checkflag
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_circfeedpolcal

Task Description

Base circfeedpolcal task
The hifv_circfeedpolcal task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
Dterm_solint -- D-terms spectral averaging
default: 2MHz
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
leakage_poltype -- string of poltype to use for override in the first polcal execution
default: '' (blank means the task heuristics will decide what poltype to use)
mbdkcross -- run gaincal KCROSS grouped by baseband
default: False
clipminmax -- Range to use for clipping
default: [0.0,0.25]
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic circfeedpolcal task
hifv_circfeedpolcal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
Dterm_solint	string	2MHz	D-terms spectral averaging
refantignore	string	None	String list of antennas to ignore
leakage_poltype	string	None	poltype to use in first polcal execution - blank string means use default heuristics
mbdkcross	bool	True	Run gaincal KCROSS grouped by baseband
clipminmax	doubleArray	0.0,0.25	Range to use for flag clipping
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_exportdata

Task Description

Prepare interferometry data for export
The hifv_exportdata task for the VLA CASA pipeline exports the data defined in the pipeline context and exports it to the data products directory, converting and or
packing it as necessary.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
exportmses -- Export the final MeasurementSets instead of the final flags,

calibration tables, and calibration instructions.
default: False
example: exportmses = True
gainmap -- The value of gainmap parameter in hifv_restoredata task put in casa_piperestorescript.py
default: False
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In 'interactive' mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in 'interactive mode'
vis -- List of visibility data files for which flagging and calibration information will be exported. Defaults to the list maintained in the pipeline context.
default: []
example: vis=['X227.ms', 'X228.ms']
session -- List of sessions one per visibility file. Currently defaults to a single virtual session containing all the visibility files in vis. In future will default to set of observing sessions defined in the context.
default: []
example: session=['session1', 'session2']
pprfile -- Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'.
default: []
example: pprfile=['PPR_GRB021004.xml']
calintents -- List of calibrator image types to be exported. Defaults to all standard calibrator intents 'BANDPASS', 'PHASE', 'FLUX'
default: ''
example: calintents='PHASE'
calimages -- List of calibrator images to be exported. Defaults to all calibrator images recorded in the pipeline context.
default: []
example: calimages=['3C454.3.bandpass', '3C279.phase']
targetimages -- List of science target images to be exported. Defaults to all science target images recorded in the pipeline context.
default: []
example: targetimages=['NGC3256.band3', 'NGC3256.band6']
products_dir -- Name of the data products subdirectory. Defaults to './'
default: ''
example: products_dir='../products'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).

default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hifv_exportdata task exports the data defined in the pipeline context
and exports it to the data products directory, converting and or packing
it as necessary.
The current version of the task exports the following products
o an XML file containing the pipeline processing request
o a tar file per ASDM / MS containing the final flags version
o a text file per ASDM / MS containing the final calibration apply list
o a FITS image for each selected calibrator source image
o a FITS image for each selected science target source image
o a tar file per session containing the caltables for that session
o a tar file containing the file web log
o a text file containing the final list of CASA commands
Issues
Support for merging the calibration state information into the pipeline
context / results structure and retrieving it still needs to be added.
Support for merging the clean results into the pipeline context / results
structure and retrieving it still needs to be added.
Support for creating the final pipeline results entity still needs to
be added.
Session information is not currently handled by the pipeline context.
By default all ASDMs are combined into one session.
Examples
1. Export the pipeline results for a single session to the data products
directory
```
!mkdir ../products
hifv_exportdata (products_dir='../products')
```
2. Export the pipeline results to the data products directory specify that
only the gain calibrator images be saved.
```
!mkdir ../products
hifv_exportdata (products_dir='../products', calintents='*PHASE*')
```

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
exportmses	bool	False	Export MeasurementSets instead of flags and caltables
pprfile	string	None	The pipeline processing request file to be exported
calintents	string	None	The calibrator source target intents to be exported
calimages	stringArray	None	List of calibrator images to be exported
targetimages	stringArray	None	List of target images to be exported
products_dir	string	None	The data products directory
gainmap	bool	False	VLASS gainmap True/False
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_exportvlassdata

Task Description

Base exportvlassdata task
The hifv_exportvlassdata task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'

--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic exportvlassdata task
hifv_exportvlassdata()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_finalcals

Task Description

Finalcals
The hifv_finalcals task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user

can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
weakbp -- Activate the weak bandpass calibrator heuristics
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Create the final calibration tables to be applied to the data in the VLA CASA pipeline.
hifv_finalcals()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
weakbp	bool	False	Activate weak bandpass heuristics
refantignore	string	None	String list of antennas to ignore

Back

---

hifv_flagbaddef

Task Description

Flagging of deformatters - amp and phase
The hifv_flagbaddef task does flagging of bad deformatters - amp and phase
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,

MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
doflagundernspwlimit -- if the number of bad spws is greater than zero and
doflagundernspwlimit is set to True, then spws are flagged individually.
default: True
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Use the bad deformatter heuristics and flag amp and phase (VLA CASA pipeline).
hifv_flagbaddef()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
doflagundernspwlimit	bool	True	If bad spws is greater than zero keyword is True, then spws are flagged individually.
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hifv_flagcal

Task Description

Base flagging task
The hifv_flagcal task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
caltable -- single string name of caltable to be flagged
clipminmax -- Range to use for clipping: default [0.9,1.1]
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Flag existing caltable
hifv_flagcal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
caltable	string	None	String name of the caltable
clipminmax	any	None	Range to use for clipping
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_flagdata

Task Description

Do basic deterministic flagging of a list of MeasurementSets
The hifv_flagdata task performs basic flagging operations on a list of MeasurementSets.
Keyword arguments:
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
autocorr -- Flag autocorrelation data.
default: True
shadow -- Flag shadowed antennas.
default: True
scan -- Flag a list of specified scans.
default: True
scannumber -- A string containing a comma delimited list of scans to be flagged.
example: '3,5,6'
default: ''
quack -- Quack scans
default: True
clip -- Clip mode
default: True
baseband -- Flag 20MHz of each edge of basebands
default: True
intents -- A string containing a comma delimited list of intents against

which the scans to be flagged are matched.
example: '*BANDPASS*'
default: '*POINTING*,*FOCUS*,*ATMOSPHERE*,*SIDEBAND_RATIO*'
edgespw -- Flag the edge spectral window channels.
default: True
fracspw -- Fraction of the baseline correlator TDM edge channels to be flagged.
default: 0.05
online -- Apply the online flags.
default: True
fileonline -- File containing the online flags. These are computed by the
h_init or hif_importdata data tasks. If the online flags files
are undefined a name of the form 'msname.flagonline.txt' is assumed.
default: ''
template -- Apply flagging templates
default: False
filetemplate -- The name of a text file that contains the flagging template
for RFI, birdies, telluric lines, etc. If the template flags files
is undefined a name of the form 'msname.flagtemplate.txt' is assumed.
default: ''
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
defined in the pipeline context.
example:
default: ''
flagbackup -- Back up any pre-existing flags.
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hifv_flagdata task performs basic flagging operations on a list of measurements including:
o autocorrelation data flagging
o shadowed antenna data flagging
o scan based flagging
o edge channel flagging
o baseband edge flagging
o applying online flags
o applying a flagging template
o quack, shadow, and basebands

o Antenna not-on-source (ANOS)
Issues
Examples
1. Do basic flagging on a MeasurementSet

   hifv_flagdata()
2. Do basic flagging on a MeasurementSet as well as flag pointing and atmosphere data
   hifv_flagdata(scan=True intent='*BANDPASS*')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets to flag
autocorr	bool	True	Flag autocorrelation data
shadow	bool	True	Flag shadowed antennas
scan	bool	True	Flag specified scans
scannumber	string	None	List of scans to be flagged
quack	bool	True	Quack scans
clip	bool	True	Clip mode
baseband	bool	True	Flag 20MHz of each edge of basebands
intents	string	*POINTING*,*FOCUS*,*ATMOSPHERE*,*SIDEBAND_RATIO*, *UNKNOWN*, *SYSTEM_CONFIGURATION*, *UNSPECIFIED#UNSPECIFIED*	List of intents of scans to be flagged
edgespw	bool	True	Flag edge channels
fracspw	double	0.05	Fraction of baseline correlator edge channels to be flagged
online	bool	True	Apply the online flags

fileonline	string	None	File of online flags to be applied
template	bool	True	Apply a flagging template
filetemplate	stringArray	None	File that contains the flagging template
hm_tbuff	string	1.5int	The time buffer computation heuristic
tbuff	any	0.0	List of time buffers (sec) to pad timerange in flag commands
pipelinemode	string	automatic	The pipeline operating mode
flagbackup	bool	False	Backup pre-existing flags before applying new ones
dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_fluxboot2

Task Description

Fluxboot2
The hifv_fluxboot2 task performs the flux density bootstrapping stage of the VLA pipeline.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
caltable -- string name of flagged caltable

fitorder -- Polynomial order of the spectral fitting for valid flux densities
with multiple spws. Currently only support 1 (spectral index only) or
2 (spectral index and curvature). It falls back to a lower fitorder if
there are not enough solutions to fit with the requested fitorder.
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. VLA CASA pipeline flux density bootstrapping.
hifv_fluxboot2()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
caltable	string	None	String name of the flagged caltable
fitorder	int	1	order of spectral fitting
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
refantignore	string	None	String list of antennas to ignore

hifv_fluxboot

Task Description

Fluxboot
The hifv_fluxboot task performs the flux density bootstrapping stage of the VLA pipeline.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
caltable -- string name of flagged caltable
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. VLA CASA pipeline flux density bootstrapping.
hifv_fluxboot()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
caltable	string	None	String name of the flagged caltable
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
refantignore	string	None	String list of antennas to ignore

Back

---

hifv_gaincurves

Task Description

Runs gencal in gc mode

The hifv_gaincurves task runs gencal in gc mode

Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode

vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs, MSs, or tar files of MSs, If ASDM files are specified, they will be converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']

caltable -- name of caltable to create

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In 'interactive' mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in 'interactive mode'

--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True

Output:

results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.

Examples

1. Load an ASDM list in the ../rawdata subdirectory into the context.

hifv_gaincurves()

Parameter List

name	**type**	**default**	**description**
vis	stringArray	None	List of input visibility data
caltable	string	None	String name of caltable
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hifv_hanning

Task Description

Hanning smoothing on a dataset
The hifv_hanning task will hanning smooth a VLA dataset
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Run the task to execute hanning smoothing on a VLA CASA pipeline loaded MeasurementSet.
hifv_hanning()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

hifv_importdata

Task Description

Imports data into the VLA pipeline
The hifv_importdata task loads the specified visibility data into the pipeline
context unpacking and / or converting it as necessary.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSes, or tar files of MSes, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
session -- List of sessions to which the visibility files belong. Defaults
to a single session containing all the visibility files, otherwise
a session must be assigned to each vis file.
default: []
example: session=['Session_1', 'Sessions_2']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
createmms -- Create a multi-MeasurementSet ('true') ready for parallel
processing, or a standard MeasurementSet ('false'). The default setting
('automatic') creates an MMS if running in a cluster environment.
default: automatic
ocorr_mode -- Read in cross- and auto-correlation data(ca), cross-
correlation data only (co), or autocorrelation data only (ao).
default: co

---- pipeline context defined parameter argument which can be set only in
'interactive mode'
asis -- SDM tables to convert as is
default: 'Receiver CalAtmosphere'
example: 'Receiver', ''
overwrite -- Overwrite existing MSes on output.
default: False
nocopy -- When importing an MS, disable copying of the MS to the working directory
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but

do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Load an ASDM list in the ../rawdata subdirectory into the context.
hifv_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e',
'../rawdata/uid_A002_x30a93d_X44e'])
2. Load an MS in the current directory into the context.
hifv_importdata (vis=[uid___A002_X30a93d_X43e.ms])
3. Load a tarred ASDM in ../rawdata into the context.
hifv_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e.tar.gz'])
4. Check the hifv_importdata inputs, then import the data
myvislist = ['uid___A002_X30a93d_X43e.ms', 'uid_A002_x30a93d_X44e.ms']
hifv_importdata(vis=myvislist, pipelinemode='getinputs')
hifv_importdata(vis=myvislist)
5. Load an ASDM but check the results before accepting them into the context.
results = hifv_importdata (vis=['uid___A002_X30a93d_X43e.ms'],
acceptresults=False)
results.accept()
6. Run in dryrun mode before running for real
results = hifv_importdata (vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)
results = hifv_importdata (vis=['uid___A002_X30a93d_X43e.ms'])

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of visibility data sessions
pipelinemode	string	automatic	The pipeline operating mode
asis	string	Receiver CalAtmosphere	ASDM to convert as is
overwrite	bool	False	Overwrite existing files on import
nocopy	bool	False	Disable copying of MS to working directory
createmms	string	automatic	Create an MMS
ocorr_mode	string	co	Default set to CROSS_ONLY (co)
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_opcal

Task Description

Runs gencal in opac mode
The hifv_opcal task runs gencal in opac mode
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
caltable -- name of caltable to create
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes

dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Load an ASDM list in the ../rawdata subdirectory into the context.
hifv_opcal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
caltable	string	None	String name of caltable
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_pbcor

Task Description

Base pbcor task
The hifv_pbcor task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.

default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic pbcor task
hifv_pbcor()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_plotsummary

Task Description

End of VLA pipeline plotsummary
The hifv_plotsummary task will generate diagnostic plots at the end of the VLA CASA pipeline run.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user

can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Execute the pipeline plotting task.
hifv_plotsummary()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_priorcals

Task Description

Runs gaincurves, opacities, requantizer gains, antenna position corrections, and tec_maps
The hifv_priorcals runs gaincurves, opacities, requantizer gains and antenna position corrections.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
swpow_spw -- Spectral-window(s) for plotting: "" ==>all, spw="6,14"
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline

determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Run gaincurves, opacities, requantizer gains and antenna position corrections.
hifv_priorcals()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
tecmaps	bool	False	Run tecmaps priorcal
swpow_spw	any	None	Spectral-window(s) for plotting: "" ==>all, spw="6,14"
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_restoredata

Task Description

Restore flagged and calibration interferometry data from a pipeline run
The hifv_restoredata task restores flagged and calibrated MeasurementSets
from archived ASDMs and pipeline flagging and calibration date products.
Keyword arguments:

---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
  determines the values of all context defined pipeline inputs automatically.
  In 'interactive' mode the user can set the pipeline context defined
  parameters manually. In 'getinputs' mode the user can check the settings
  of all pipeline parameters without running the task.
  default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of raw visibility data files to be restored. Assumed to be
  in the directory specified by rawdata_dir.
  default: None
  example: vis=['uid___A002_X30a93d_X43e']
session -- List of sessions one per visibility file.
  default: []
  example: session=['session_3']
products_dir -- Name of the data products directory. Currently not
  used.
  default: '../products'
  example: products_dir='myproductspath'
copytoraw -- Copy calibration and flagging tables to rawdata directory
  default: True
rawdata_dir -- Name of the rawdata subdirectory.
  default: '../rawdata'
  example: rawdata_dir='myrawdatapath'
lazy -- Use the lazy filler option
  default: False
  example: lazy=True
bdfflags -- Set the BDF flags
  default: True
  example: bdfflags=False
ocorr_mode -- Set ocorr_mode
  default: 'ca'
  example: ocorr_mode='ca'
gainmap -- If True, map gainfields to a particular list of scans when
  applying calibration tables
  default: False
asis -- Set list of tables to import as is
  default: ''
  example: ocorr_mode='Source Receiver'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
  do not execute (False).
  default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
  reject them (False).
  default: True

Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hifv_restoredata restores flagged and calibrated data from archived
ASDMs and pipeline flagging and calibration data products. Pending archive
retrieval support hifv_restoredata assumes that the required products
are available in the rawdata_dir in the format produced by the
hifv_exportdata task.
hifv_restoredata assumes that the following entities are available in the raw
data directory
o the ASDMs to be restored
o for each ASDM in the input list
  o a compressed tar file of the final flagversions file, e.g.
    uid___A002_X30a93d_X43e.ms.flagversions.tar.gz
  o a text file containing the applycal instructions, e.g.
    uid___A002_X30a93d_X43e.ms.calapply.txt
  o a compressed tar file containing the caltables for the parent session,
    e.g. uid___A001_X74_X29.session_3.caltables.tar.gz
hifv_restoredata performs the following operations
o imports the ASDM(s))
o removes the default MS.flagversions directory created by the filler
o restores the final MS.flagversions directory stored by the pipeline
o restores the final set of pipeline flags to the MS
o restores the final calibration state of the MS
o restores the final calibration tables for each MS
o applies the calibration tables to each MS
Issues
Examples
1. Restore the pipeline results for a single ASDM in a single session
  hifv_restoredata (vis=['myVLAsdm'], session=['session_1'], ocorr_mode='ca')

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
products_dir	string	../products	The archived pipeline data products directory
copytoraw	bool	True	Copy calibration and flagging tables to raw data directory
rawdata_dir	string	../rawdata	The rawdata directory
lazy	bool	False	Use the lazy filler option
bdfflags	bool	False	Set the BDF flags
ocorr_mode	string	co	Correlation import mode
gainmap	bool	False	VLASS gainmap True/False
asis	string	None	List of tables to import asis
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_rqcal

Task Description

Runs gencal in rq mode
The hifv_rqcal task runs gencal in rq mode for Requantizer Gains
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
caltable -- name of caltable to create
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.

---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Load an ASDM list in the ../rawdata subdirectory into the context.
hifv_rqcal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
caltable	string	None	String name of caltable
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_semiFinalBPdcals

Task Description

Runs a second quick calibration to set up for heuristic flagging
The hifv_semiFinalBPdcals task runs a second quick calibration to set up for heuristic flagging:
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
weakbp -- Activate the weak bandpass calibrator heuristics
default: True
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline

context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Heuristic flagging
hifv_semiFinalBPdcals()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
weakbp	bool	False	Activate weak bandpass heuristics
refantignore	string	None	String list of antennas to ignore

Back

hifv_solint

Task Description

Determines different solution intervals
The hifv_solint task determines different solution intervals
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
limit_short_solint -- keyword argument in units of seconds to limit the short solution interval.
Can be a string or float numerical value in units of seconds of '0.45' or 0.45.
Can be set to a string value of 'int'.
default: ''
refantignore -- string list to be ignored as reference antennae.
default: ''
Example: refantignore='ea02,ea03'
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
limit_short_solint -- string keyword argument in units of seconds to limit the short solution interval.
Can be set to a string value of 'int'.
default: ''
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Determines different solution intervals:

hifv_solint()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
limit_short_solint	any	None	Limiting solint value
refantignore	string	None	String list of antennas to ignore

Back

---

hifv_statwt

Task Description

Statwt
The hifv_statwt task performs statistical weighting of the visibilities
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).

default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Statistical weighting of the visibilities:
hifv_statwt()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_swpowcal

Task Description

Runs gencal in swpow mode
The hifv_swpowcal task runs gencal in swpow mode for Switched Power
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
caltable -- name of caltable to create
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).

default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Load an ASDM list in the ../rawdata subdirectory into the context.
hifv_swpowcal()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
caltable	string	None	String name of caltable
spw	any	None	Spectral-window/frequency/channel: '' ==> all, spw="0:17~19"
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_syspower

Task Description

Base syspower task
The hifv_syspower task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
clip_sp_template -- clipping range
default: [0.7, 1.2]
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'

vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic syspower task
hifv_syspower()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
clip_sp_template	doubleArray	0.7, 1.2	Clipping range
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifv_targetflag

Task Description

Targetflag
The hifv_targetflag task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Run rflag on both the science targets and calibrators:
hifv_targetflag()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
intents	string	*CALIBRATE*,*TARGET*	List of intents of scans to be flagged

Back

---

hifv_tecmaps

Task Description

Base tecmaps task
The hifv_tecmaps task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode

vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Basic tecmaps task
hifv_tecmaps()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hifv_testBPdcals

Task Description

Runs initial delay calibration to set up heuristic flagging
The hifv_testBPdcals task does an initial delay calibration to set up
heuristic flagging.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
refantignore -- string list to be ignored as reference antennas.
default: ''
Example: refantignore='ea02,ea03'
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
weakbp -- Activate the weak bandpass calibrator heuristics
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Initial delay calibration to set up heuristic flagging.
hifv_testBPdcals()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
weakbp	bool	False	Activate weak bandpass heuristics
refantignore	string	None	String list of antennas to ignore

Back

---

hifv_vlasetjy

Task Description

Does an initial setjy run on the vis
The hifv_vlasetjy task does an initial run of setjy on the vis
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs,
MSs, or tar files of MSs, If ASDM files are specified, they will be
converted to MS format.
default: []
example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

Examples
1. Initial run of setjy:
   hifv_vlasetjy()

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
field	string	None	List of field names or ids
intent	string	None	Observing intent of flux calibrators
spw	string	None	List of spectral window ids
model	string	None	File location for field model
reffile	string	None	Path to file with fluxes for non-solar system calibrators
fluxdensity	any	-1	Specified flux density [I,Q,U,V]; -1 will lookup values
spix	double	0.0	Spectral index of fluxdensity
reffreq	string	1GHz	Reference frequency for spix
scalebychan	bool	True	Scale the flux density on a per channel basis or else on a per spw basis
standard	variant	None	Flux density standard
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

Created at Fri Sep 14 10:04:26 2018 UTC
This page is automatically generated from XML interface definition files.

Summary of single-dish tasks and parameters

hsd_applycal

Task Description

Apply the calibration(s) to the data
Apply precomputed calibrations to the data.
---- pipeline parameter arguments which can be set in any pipeline mode
applymode -- Calibration apply mode
''='calflagstrict': calibrate data and apply flags from solutions using
the strict flagging convention
'trial': report on flags from solutions, dataset entirely unchanged
'flagonly': apply flags from solutions only, data not calibrated
'calonly': calibrate data only, flags from solutions NOT applied
'calflagstrict':
'flagonlystrict': same as above except flag spws for which calibration is
unavailable in one or more tables (instead of allowing them to pass
uncalibrated and unflagged)
default: ''
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
in the pipeline context.
default: []
example: ['X227.ms']
field -- A string containing the list of field names or field ids to which
the calibration will be applied. Defaults to all fields in the pipeline
context.
default: ''
example: '3C279', '3C279, M82'
intent -- A string containing the list of intents against which the
selected fields will be matched. Defaults to all supported intents
in the pipeline context.
default: ''
example: '*TARGET*'
spw -- The list of spectral windows and channels to which the calibration
will be applied. Defaults to all science windows in the pipeline
context.
default: ''
example: '17', '11, 15'
antenna -- The list of antennas to which the calibration will be applied.
Defaults to all antennas. Not currently supported.

--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
  do not execute (False).
  default: False
acceptresults -- Add the results of the task to the pipeline context (True) or
  reject them (False).
  default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
   the results object for the pipeline task is returned
Description
hif_applycal applies the precomputed calibration tables stored in the pipeline
context to the set of visibility files using predetermined field and
spectral window maps and default values for the interpolation schemes.
Users can interact with the pipeline calibration state using the tasks
hif_export_calstate and hif_import_calstate.
Issues
There is some discussion about the appropriate values of calwt. Given
properly scaled data, the correct value should be the CASA default of True.
However at the current time ALMA is suggesting that calwt be set to True for
applying observatory calibrations, e.g. antenna positions, WVR, and system
temperature corrections, and to False for applying instrument calibrations,
e.g. bandpass, gain, and flux.
Examples
1. Apply the calibration to the target data
hsd_applycal (intent='TARGET')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets
field	string	None	Set of data selection field names or ids
intent	string	None	Set of data selection observing intents
spw	string	None	Set of data selection spectral window/channels
antenna	string	None	Set of data selection antenna ids
applymode	string	None	Calibration mode: ""="calflagstrict","calflag","calflagstrict","trial","flagonly","flagonlystrict", or "calonly"
calwt	boolArray	True	Calibrate the weights as well as the data
flagbackup	bool	True	Backup the flags before the apply
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run task (False) or display the command(True)
acceptresults	bool	True	Automatically accept results into the context

Back

---

hsd_baseline

Task Description

Detect and validate spectral lines, subtract baseline by masking detected lines
The hsd_baseline task subtracts baseline from calibrated spectra.
By default, the task tries to find spectral line feature using
line detection and validation algorithms. Then, the task puts a
mask on detected lines and perform baseline subtraction. The user
is able to turn off automatic line masking by setting linewindow
parameter, which specifies pre-defined line window.
Fitting order is automatically determined by default. It can be
disabled by specifying fitorder as non-negative value. In this case,
the value specified by fitorder will be used.
WARNING
Currently, hsd_baseline overwrites the result obtained by the previous
run. Due to this behavior, users need to be careful about an order
of the task execution when they run hsd_baseline multiple times with
different data selection. Suppose there are two spectral windows
(0 and 1) and hsd_baseline is executed separately for each spw as
below,
hsd_baseline(pipelinemode="interactive", spw="0")
hsd_baseline(pipelinemode="interactive", spw="1")
hsd_blflag(pipelinemode="automatic")

hsd_imaging(pipelinemode="automatic")
Since the second run of hsd_baseline overwrites the result for spw 0
with the data before baseline subtraction, this will not produce
correct result for spw 0. Proper sequence for this use case is to
process each spw to the imaging stage separately, which looks like
as follows:
hsd_baseline(pipelinemode="interactive", spw="0")
hsd_blflag(pipelinemode="interactive", spw="0")
hsd_imaging(pipelinemode="interactive", spw="0"))
hsd_baseline(pipelinemode="interactive", spw="1")
hsd_blflag(pipelinemode="interactive", spw="1")
hsd_imaging(pipelinemode="interactive", spw="1")

Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
fitfunc -- fitting function for baseline subtraction. You can only choose
  cubic spline ('spline' or 'cspline')
  default: 'cspline'.
fitorder -- Fitting order for polynomial. For cubic spline, it is used
  to determine how much the spectrum is segmented into. Default (-1) is
  to determine the order automatically.
  default: -1 (auto determination)
linewindow -- Pre-defined line window. If this is set, specified line
  windows are used as a line mask for baseline subtraction instead to
  determine masks based on line detection and validation stage. Several
  types of format are acceptable. One is channel-based window,

    [min_chan, max_chan]

  where min_chan and max_chan should be an integer. For multiple
  windows, nested list is also acceptable,

    [[min_chan0, max_chan0], [min_chan1, max_chan1], ...]

  Another way is frequency-based window,

    [min_freq, max_freq]

  where min_freq and max_freq should be either a float or a string.
  If float value is given, it is interpreted as a frequency in Hz.
  String should be a quantity consisting of "value" and "unit", e.g.,
  '100GHz'. Multiple windows are also supported.

    [[min_freq0, max_freq0], [min_freq1, max_freq1], ...]

  Note that the specified frequencies are assumed to be the value in
  LSRK frame. Note also that there is a limitation when multiple MSes are

processed. If native frequency frame of the data is not LSRK (e.g. TOPO), frequencies need to be converted to that frame. As a result, corresponding channel range may vary between MSes. However, current implementation is not able to handle such case. Frequencies are converted to desired frame using representative MS (time, position, direction).

In the above cases, specified line windows are applied to all science spws. In case when line windows vary with spw, line windows can be specified by a dictionary whose key is spw id while value is line window. For example, the following dictionary gives different line windows to spws 17 and 19. Other spws, if available, will have an empty line window.

{17: [[100, 200], [1200, 1400]], 19: ['112115MHz', '112116MHz']}

Furthermore, linewindow accepts MS selection string. The following string gives [[100,200],[1200,1400]] for spw 17 while [1000,1500] for spw 21.

"17:100~200;1200~1400,21:1000~1500"

The string also accepts frequency with units. Note, however, that frequency reference frame in this case is not fixed to LSRK. Instead, the frame will be taken from the MS (typically TOPO for ALMA). Thus, the following two frequency-based line windows result different channel selections.

{19: ['112115MHz', '112116MHz']} # frequency frame is LSRK
"19:11215MHz~11216MHz" # frequency frame is taken from the data
(TOPO for ALMA)

default: [] (do line detection and validation)
example: [100,200] (channel), [115e9, 115.1e9] (frequency in Hz)
['115GHz', '115.1GHz'], see above for more examples

linewindowmode -- Merge or replace given manual line window with line detection/validation result. If 'replace' is given, line detection and validation will not be performed. On the other hand, when 'merge' is specified, line detection/validation will be performed and manually specified line windows are added to the result. Note that this has no effect when linewindow for target spw is empty. In that case, line detection/validation will be performed regardless of the value of linewindowmode.
default: 'replace'
options: 'replace', 'merge'

edge -- number of edge channels to be dropped from baseline subtraction. the value must be a list with length of 2, whose values specifies left and right edge channels respectively.

default: [] ([0,0])
example: [10,10]
broadline -- Try to detect broad component of spectral line if True.
default: True
clusteringalgorithm -- selection of the algorithm used in the clustering
analysis to check the validity of detected line features. 'kmean'
algorithm and hierarchical clustering algorithm 'hierarchy' are so
far implemented.
default: 'kmean'.

deviationmask -- Apply deviation mask in addition to masks determined by
the automatic line detection.
default: True
pipelinemode -- The pipeline operating mode. In 'automatic' mode the
pipeline determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
infiles -- List of data files. These must be a name of MeasurementSets that
are registered to context via hsd_importdata task.
default: []
example: vis=['X227.ms', 'X228.ms']

field -- Data selection by field.
default: '' (all fields)
example: '1' (select by FIELD_ID)
'M100*' (select by field name)
antenna -- Data selection by antenna.
default: '' (all antennas)
example: '1' (select by ANTENNA_ID)
'PM03' (select by antenna name)
spw -- Data selection by spw.
default: '' (all spws)
example: '3,4' (generate caltable for spw 3 and 4)
['0','2'] (spw 0 for first data, 2 for second)
pol -- Data selection by polarizations.
default: '' (all polarizations)
example: '0' (generate caltable for pol 0)
['0~1','0'] (pol 0 and 1 for first data, only 0 for second)
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples

Parameter List

name	**type**	**default**	**description**
fitfunc	string	cspline	Fitting function for baseline subtraction
fitorder	int	-1	Fitting order for baseline subtraction
linewindow	any	None	Pre-defined line window
linewindowmode	string	replace	Merge or replace given manual line window with line detection/validation result
edge	intArray	None	Edge channels to be dropped
broadline	bool	True	Try to detect broad component of the line
clusteringalgorithm	string	kmean	Algorithm for line validation clustering algorithm
deviationmask	bool	True	Apply deviation mask in addition to detected line masks
pipelinemode	string	automatic	The pipeline operating mode
infiles	stringArray	None	List of input files to be calibrated (default all)
field	string	None	select data by field
antenna	string	None	select data by antenna
spw	any	None	select data by spw ids, e.g. \'3,5,7\' (\'\'=all)
pol	any	None	select data by polarizations, e.g. \'0~1\' (\'\'=all)
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context
parallel	string	automatic	Execute using CASA HPC functionality, if available.

Back

hsd_blflag

Task Description

Flag spectra based on predefined criteria of single dish pipeline
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
iteration -- Number of iterations to perform sigma clipping to calculate
threshold value of flagging.
default: 5
edge -- Number of channels to be dropped from the edge.
The value must be a list of integer with length of one or two.
If list length is one, same number will be applied both side of
the band.
default: [0,0]
example: [10,20], [10]
flag_tsys -- Activate (True) or deactivate (False) Tsys flag.
default: True
tsys_thresh -- Threshold value for Tsys flag.
default: 3.0
flag_weath -- Activate (True) or deactivate (False) weather flag.
Since weather flagging is not implemented yet, setting True
has no effect at the moment.
default: False
weath_thresh -- Threshold value for weather flag.
default: 3.0
flag_prfre -- Activate (True) or deactivate (False) flag by expected rms
of pre-fit spectra.
default: True
prfre_thresh -- Threshold value for flag by expected rms of pre-fit
spectra.
default: 3.0
flag_pofre -- Activate (True) or deactivate (False) flag by expected rms
of post-fit spectra.
default: True
pofre_thresh -- Threshold value for flag by expected rms of post-fit
spectra.
default: 1.3333
flag_prfr -- Activate (True) or deactivate (False) flag by rms of pre-fit
spectra.
default: True
prfr_thresh -- Threshold value for flag by rms of pre-fit spectra.
default: 4.5
flag_pofr -- Activate (True) or deactivate (False) flag by rms of post-fit
spectra.
default: True
pofr_thresh -- Threshold value for flag by rms of post-fit spectra.

default: 4.0
flag_prfrm -- Activate (True) or deactivate (False) flag by running mean
of pre-fit spectra.
default: True
prfrm_thresh -- Threshold value for flag by running mean of pre-fit
spectra.
default: 5.5
prfrm_nmean -- Number of channels for running mean of pre-fit spectra.
default: 5
flag_pofrm -- Activate (True) or deactivate (False) flag by running mean
of post-fit spectra.
default: True
pofrm_thresh -- Threshold value for flag by running mean of post-fit
spectra.
default: 5.0
pofrm_nmean -- Number of channels for running mean of post-fit spectra.
default: 5
flag_user -- Activate (True) or deactivate (False) user-defined flag.
Since user flagging is not implemented yet, setting True
has no effect at the moment.
default: False
user_thresh -- Threshold value for flag by user-defined rule.
default: 3.0
plotflag -- True to plot result of data flagging.
default: True
pipelinemode -- The pipeline operating mode. In 'automatic' mode the
pipeline determines the values of all context defined pipeline
inputs automatically. In interactive mode the user can set the
pipeline context defined parameters manually. In 'getinputs' mode
the user can check the settings of all pipeline parameters without
running the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode' or 'getinputs' modes

infiles -- ASDM or MS files to be processed. This parameter behaves as
data selection parameter. The name specified by infiles must be
registered to context before you run hsd_blflag.
default: [] (process all data in context)
antenna -- Data selection by antenna names or ids.
default: '' (all antennas)
example: 'PM03,PM04'

field -- Data selection by field names or ids.
default: '' (all fields)
example: '*Sgr*,M100'

spw -- Data selection by spw ids.
  default: '' (all spws)
  example: '3,4' (spw 3 and 4)
pol -- Data selection by polarizations.
  default: '' (all polarizations)
  example: 'XX,YY' (correlation XX and YY)
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
        do not execute (False).
    default: True
acceptresults -- Add the results of the task to the pipeline context
    (True) or reject them (False).
    default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
    the results object for the pipeline task is returned.
Description
The hsd_blflag
6. Flagging
Data are flagged based on several flagging rules. Available rules are:
expected rms, calculated rms, and running mean of both pre-fit and
post-fit spectra. Tsys flagging is also available.
In addition, the heuristics script creates many plots for each stage.
Those plots are included in the weblog.
Issues
Examples

Parameter List

name	type	default	description
iteration	int	5	Number of iteration to perform sigma clipping to calculate threshold
edge	intArray	0,0	Number of edge channels to be excluded from statistic calculation to flag data
flag_tsys	bool	True	Flag data by Tsys value
tsys_thresh	double	3.0	Threshold for Tsys flag
flag_weath	bool	False	Flag data by weather (not implemented yet)
weath_thresh	double	3.0	Threshold for weather flag
flag_prfre	bool	True	Flag data by EXPECTED RMS of pre-fit spectra
prfre_thresh	double	3.0	Threshold for EXPECTED RMS of pre-fit spectra flag

flag_pofre	bool	True	Flag data by EXPECTED RMS of post-fit spectra
pofre_thresh	double	1.3333	Threshold for EXPECTED RMS of post-fit spectra flag
flag_prfr	bool	True	Flag data by RMS of pre-fit spectra
prfr_thresh	double	4.5	Threshold for RMS of pre-fit flag
flag_pofr	bool	True	Flag data by RMS of post-fit spectra
pofr_thresh	double	4.0	Threshold for RMS of post-fit spectra flag
flag_prfrm	bool	True	Flag data by running mean of pre-fit spectra
prfrm_thresh	double	5.5	Threshold for running mean of pre-fit spectra flag
prfrm_nmean	int	5	Number of channels for running mean of pre-fit spectra flag
flag_pofrm	bool	True	Flag data by running mean of post-fit spectra
pofrm_thresh	double	5.0	Threshold for running mean of post-fit spectra flag
pofrm_nmean	int	5	Number of channels for running mean of post-fit spectra flag
flag_user	bool	False	Flag data by user flag (not implemented yet)
user_thresh	double	5.0	Threshold for user flag
plotflag	bool	True	Create plots for flagging
pipelinemode	string	automatic	The pipeline operating mode
infiles	stringArray	None	List of input files to be flagged (\'\'=all)
antenna	string	None	select data by antenna names or ids, e.g. \'PM03,PM04\' (\'\'=all)
field	string	None	select data by field names or ids, e.g. \'M100,Sgr*\' (\'\'=all)
spw	string	None	select data by spectral windows, e.g. \'3,5,7\' (\'\'=all)
pol	string	None	select data by polarizations, e.g. \'XX,YY\' (\'\'=all)
dryrun	bool	False	Run the task (False) or display the task command (True)
acceptresults	bool	True	Add the results into the pipeline context
parallel	string	automatic	Execute using CASA HPC functionality, if available.

Back

hsd_exportdata

Task Description

Prepare single dish data for export
The hsd_exportdata task exports the data defined in the pipeline context
and exports it to the data products directory, converting and or
packing it as necessary.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In 'interactive' mode the user can set the pipeline context defined
parameters manually. In 'getinputs' mode the user can check the settings
of all pipeline parameters without running the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
pprfile -- Name of the pipeline processing request to be exported. Defaults
to a file matching the template 'PPR_*.xml'.
default: []
example: pprfile=['PPR_GRB021004.xml']
targetimages -- List of science target images to be exported. Defaults to all
science target images recorded in the pipeline context.
default: []
example: targetimages=['r_aqr.CM02.spw5.line0.XXYY.sd.im',
'r_aqr.CM02.spw5.XXYY.sd.cont.im']
products_dir -- Name of the data products subdirectory. Defaults to './'
default: ''
example: products_dir='../products'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hsd_exportdata task exports the data defined in the pipeline context
and exports it to the data products directory, converting and or packing
it as necessary.
The current version of the task exports the following products
o a FITS image for each selected science target source image
o a tar file per ASDM containing the final flags version and blparam

o a tar file containing the file web log
TBD
o a file containing the line feature table(frequency, width, spatial distribution)
o a file containing the list of identified transitions from line catalogs
Examples
1. Export the pipeline results for a single session to the data products
directory
!mkdir ../products
hsd_exportdata (products_dir='../products')

Parameter List

name	type	default	description
pprfile	string	None	The pipeline processing request (PPR) file to be exported
targetimages	stringArray	None	List of target CASA images to be exported
products_dir	string	None	The data products directory
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hsd_flagdata

Task Description

Do basic flagging of a list of MeasurementSets
The hsd_flagdata data performs basic flagging operations on a list of
MeasurementSets.
Keyword arguments:
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs automatically.
In interactive mode the user can set the pipeline context defined parameters
manually. In 'getinputs' mode the user can check the settings of all
pipeline parameters without running the task.
default: 'automatic'.
---- pipeline parameter arguments which can be set in any pipeline mode
autocorr -- Flag autocorrelation data.
default: False
shadow -- Flag shadowed antennas.
default: True
scan -- Flag a list of specified scans.

default: True
scannumber -- A string containing a comma delimited list of scans to be
flagged.
example: '3,5,6'
default: ''
intents -- A string containing a comma delimited list of intents against
which the scans to be flagged are matched.
example: '*BANDPASS*'
default: 'POINTING,FOCUS,ATMOSPHERE,SIDEBAND'
edgespw -- Flag the edge spectral window channels.
default: True
fracspw -- Fraction of the baseline correlator TDM edge channels to be flagged.
default: 0.0625
fracspwfps -- Fraction of the ACS correlator TDM edge channels to be flagged.
default: 0.48387
online -- Apply the online flags.
default: True
fileonline -- File containing the online flags. These are computed by the
h_init or hif_importdata data tasks. If the online flags files
are undefined a name of the form 'msname.flagonline.txt' is assumed.
default: ''
template -- Apply flagging templates
default: True
filetemplate -- The name of a text file that contains the flagging template
for RFI, birdies, telluric lines, etc. If the template flags files
is undefined a name of the form 'msname.flagtemplate.txt' is assumed.
default: ''
hm_tbuff -- The heuristic for computing the default time interval padding parameter.
The options are 'halfint' and 'manual'. In 'halfint' mode tbuff is set to
half the maximum of the median integration time of the science and calibrator target
observations.
default: 'halfint'
tbuff -- The time in seconds used to pad flagging command time intervals if
hm_tbuff='manual'.
default: 0.0
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
vis -- The list of input MeasurementSets. Defaults to the list of MeasurementSets
defined in the pipeline context.
example:
default: ''
flagbackup -- Back up any pre-existing flags.
default: False
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True

acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
The hsd_flagdata task performs basic flagging operations on a list of
MeasurementSets including:
o applying online flags
o applying a flagging template
o shadowed antenna data flagging
o scan-based flagging by intent or scan number
o edge channel flagging
Issues
Examples
1. Do basic flagging on a MeasurementSet

hsd_flagdata()
2. Do basic flagging on a MeasurementSet flagging additional scans selected
by number as well.
hsd_flagdata(scannumber='13,18')

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets to flag
autocorr	bool	False	Flag autocorrelation data
shadow	bool	True	Flag shadowed antennas
scan	bool	True	Flag specified scans
scannumber	string	None	List of scans to be flagged
intents	string	POINTING,FOCUS,ATMOSPHERE,SIDEBAND,CHECK	List of intents of scans to be flagged

edgespw	bool	True	Flag edge channels
fracspw	any	1.875GHz	Fraction of baseline correlator edge channels to be flagged
fracspwfps	double	0.048387	Fraction of ACA correlator edge channels to be flagged
online	bool	True	Apply the online flags
fileonline	string	None	File of online flags to be applied
template	bool	True	Apply a flagging template
filetemplate	stringArray	None	File that contains the flagging template
hm_tbuff	string	halfint	The time buffer computation heuristic
tbuff	any	0.0	List of time buffers (sec) to pad timerange in flag commands
qa0	bool	True	QA0 flags
qa2	bool	True	QA2 flags
pipelinemode	string	automatic	The pipeline operating mode
flagbackup	bool	False	Backup pre-existing flags before applying new ones

dryrun	bool	False	Run the task (False) or display the command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hsd_imaging

Task Description

Generate single dish images
The hsd_imaging task generates single dish images per antenna as well as combined image over whole antennas for each field and spectral window. Image configuration (grid size, number of pixels, etc.) is automatically determined based on meta data such as antenna diameter, map extent, etc. Note that generated images are always in LSRK frame.

Keyword arguments:
mode -- imaging mode controls imaging parameters in the task.
  Accepts either 'line' (spectral line imaging) or 'ampcal' (image settings for amplitude calibrator)
  default: 'line'
  options: 'line', 'ampcal',
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In 'interactive' mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
  default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in 'interactive mode'
infiles -- List of data files. These must be a name of MeasurementSets that are registered to context via hsd_importdata task.
  default: []
  example: vis=['uid___A002_X85c183_X36f.ms', 'uid___A002_X85c183_X60b.ms']
field -- Data selection by field names or ids.
  default: '' (all fields)
  example: '*Sgr*,M100'

spw -- Data selection by spw ids.

default: '' (all spws)
example: '3,4' (generate images for spw 3 and 4)
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise the results object for the pipeline task is returned.
Examples

Parameter List

name	type	default	description
mode	string	line	Imaging mode [\'line\'|\'ampcal\']
restfreq	stringArray	None	Rest frequency
pipelinemode	string	automatic	The pipeline operating mode
infiles	stringArray	None	List of input files (default all)
field	string	None	Field to be imaged, e.g., \'M100,Sgr*\' (default all)
spw	any	None	select data by spectral window IDs, e.g. \'3,5,7\' (default all)
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hsd_importdata

Task Description

Imports data into the single dish pipeline
The hsd_importdata task loads the specified visibility data into the pipeline context unpacking and / or converting it as necessary.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs, MSes, or tar files of MSes, If ASDM files are specified, they will be converted to MS format.
default: []

example: vis=['X227.ms', 'asdms.tar.gz']
session -- List of sessions to which the visibility files belong. Defaults
to a single session containing all the visibility files, otherwise
a session must be assigned to each vis file.
default: []
example: session=['Session_1', 'Sessions_2']
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
determines the values of all context defined pipeline inputs
automatically. In 'interactive' mode the user can set the pipeline
context defined parameters manually. In 'getinputs' mode the user
can check the settings of all pipeline parameters without running
the task.
default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
asis -- ASDM tables to convert as is
default: 'Antenna Station Receiver CalAtmosphere'
example: 'Receiver', ''
process_caldevice -- Ingest the ASDM caldevice table
default: False
example: True
overwrite -- Overwrite existing MSes on output.
default: False
bdfflags -- Apply BDF flags on line
default: True

lazy -- Use the lazy filter import
default: False
with_pointing_correction -- add (ASDM::Pointing::encoder - ASDM::Pointing::pointingDirection)
to the value to be written in MS::Pointing::direction
default: True
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Load an ASDM list in the ../rawdata subdirectory into the context.
hsd_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e',
'../rawdata/uid_A002_x30a93d_X44e'])
2. Load an MS in the current directory into the context.
hsd_importdata (vis=['uid___A002_X30a93d_X43e.ms'])

3. Load a tarred ASDM in ../rawdata into the context.
   hsd_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e.tar.gz'])
4. Check the hsd_importdata inputs, then import the data
   myvislist = ['uid___A002_X30a93d_X43e.ms', 'uid_A002_x30a93d_X44e.ms']
   hsd_importdata(vis=myvislist, pipelinemode='getinputs')
   hsd_importdata(vis=myvislist)
5. Load an ASDM but check the results before accepting them into the context.
   results = hsd_importdata (vis=['uid___A002_X30a93d_X43e.ms'],
      acceptresults=False)
   results.accept()
6. Run in dryrun mode before running for real
   results = hsd_importdata (vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)
   results = hsd_importdata (vis=['uid___A002_X30a93d_X43e.ms'])

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of visibility data sessions
pipelinemode	string	automatic	The pipeline operating mode
asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR	ASDM to convert as is
process_caldevice	bool	False	Import the caldevice table from the ASDM
overwrite	bool	False	Overwrite existing files on import
nocopy	bool	False	Disable copying of MS to working directory
bdfflags	bool	True	Apply BDF flags on import
lazy	bool	False	Use the lazy import option
with_pointing_correction	bool	True	Apply pointing correction to DIRECTION
createmms	string	false	Create an MMS
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hsd_k2jycal

Task Description

Derive Kelvin to Jy calibration tables

Derive the Kelvin to Jy calibration for list of MeasurementSets.

Keyword arguments

pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. In interactive mode the user can set the pipeline context defined parameters manually. In 'getinputs' mode the user can check the settings of all pipeline parameters without running the task.
default: 'automatic'.

reffile -- Path to a file containing Jy/K factors for science data, which must be provided by associating calibrator reduction or the observatory measurements. Jy/K factor must take into account all efficiencies, i.e., it must be a direct conversion factor from Ta* to Jy. The file must be in either MS-based or session-based format. The MS-based format must be in an CSV format with five fields: MS name, antenna name, spectral window id, polarization string, and Jy/K conversion factor. Example for the file is as follows:

```
MS,Antenna,Spwid,Polarization,Factor
uid___A002_X316307_X6f.ms,CM03,5,XX,10.0
uid___A002_X316307_X6f.ms,CM03,5,YY,12.0
uid___A002_X316307_X6f.ms,PM04,5,XX,2.0
uid___A002_X316307_X6f.ms,PM04,5,YY,5.0
```

The first line in the above example is a header which may or may not exist. Example for the session-based format is as follows:

```
#OUSID=XXXXXX
#OBJECT=Uranus
#FLUXJY=yy,zz,aa
#FLUXFREQ=YY,ZZ,AA
#sessionID,ObservationStartDate(UTC),ObservationEndDate(UTC),Antenna,BandCenter(MHz),BandWidth(MHz),POL,Factor
1,2011-11-11 01:00:00,2011-11-11 01:30:00,CM02,86243.0,500.0,I,10.0
1,2011-11-11 01:00:00,2011-11-11 01:30:00,CM02,86243.0,1000.0,I,30.0
1,2011-11-11 01:00:00,2011-11-11 01:30:00,CM03,86243.0,500.0,I,50.0
1,2011-11-11 01:00:00,2011-11-11 01:30:00,CM03,86243.0,1000.0,I,70.0
1,2011-11-11 01:00:00,2011-11-11 01:30:00,ANONYMOUS,86243.0,500.0,I,30.0
1,2011-11-11 01:00:00,2011-11-11 01:30:00,ANONYMOUS,86243.0,1000.0,I,50.0
2,2011-11-13 01:45:00,2011-11-13 02:15:00,PM04,86243.0,500.0,I,90.0
2,2011-11-13 01:45:00,2011-11-13 02:15:00,PM04,86243.0,1000.0,I,110.0
2,2011-11-13 01:45:00,2011-11-13 02:15:00,ANONYMOUS,86243.0,500.0,I,90.0
2,2011-11-13 01:45:00,2011-11-13 02:15:00,ANONYMOUS,86243.0,1000.0,I,110.0
```

The line starting with '#' indicates a meta data section and header.

The header must exist. The factor to apply is identified by matching the
session ID, antenna name, frequency and polarization of data in each line of
the file. Note the observation date is supplementary information and not used
for the matching so far. The lines whose antenna name is 'ANONYMOUS' are used
when there is no measurement for specific antenna in the session. In the above
example, if science observation of session 1 contains the antenna PM04, Jy/K
factor for ANONYMOUS antenna will be applied since there is no measurement for
PM04 in session 1.
If no file name is specified or specified file doesn't exist, all Jy/K factors
are set to 1.0.
default: 'jyperk.csv'
example: reffile='', reffile='working/jyperk.csv'
---- pipeline parameter arguments which can be set in any pipeline mode
---- pipeline context defined parameter arguments which can be set only in
'interactive mode'
infiles -- List of input MeasurementSets.
default: none
example: vis='ngc5921.ms'
caltable -- Name of output gain calibration tables.
default: none
example: caltable='ngc5921.gcal'
-- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Description
Derive the Kelvin to Jy calibration for list of MeasurementSets.
Issues
Example
1. Compute the Kevin to Jy calibration tables for a list of MeasurementSets:
hsd_k2jycal()

Parameter List

name	type	default	description
reffile	string	jyperk.csv	File of Jy/K conversion factor
pipelinemode	string	automatic	The pipeline operations mode
infiles	stringArray	None	List of input MeasurementSets
caltable	stringArray	None	List of output caltable(s)
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

---

hsd_restoredata

Task Description

Restore flagged and calibration single dish data from a pipeline run
The hsd_restoredata task restores flagged and calibrated MeasurementSets
from archived ASDMs and pipeline flagging and calibration date products.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
  determines the values of all context defined pipeline inputs automatically.
  In 'interactive' mode the user can set the pipeline context defined
  parameters manually. In 'getinputs' mode the user can check the settings
  of all pipeline parameters without running the task.
  default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
vis -- List of raw visibility data files to be restored. Assumed to be
  in the directory specified by rawdata_dir.
  default: None
  example: vis=['uid___A002_X30a93d_X43e']
session -- List of sessions one per visibility file.
  default: []
  example: session=['session_3']
products_dir -- Name of the data products directory. Currently not
  used.
  default: '../products'
  example: products_dir='myproductspath'

copytoraw -- Copy calibration and flagging tables to raw data directory.
  default: True
  example: copytoraw=False

rawdata_dir -- Name of the raw data directory.
  default: '../rawdata'
  example: rawdata_dir='myrawdatapath'
lazy -- Use the lazy filler option
  default: False
  example: lazy=True
bdfflags -- Set the BDF flags
  default: True
  example: bdfflags=False
ocorr_mode -- Set ocorr_mode
  default: 'ca'
  example: ocorr_mode='ca'
asis -- Set list of tables to import as is
  default: 'Antenna Station Receiver Source CalAtmosphere CalWVR'
  example: asis='Source Receiver'
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
  do not execute (False).
  default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
  reject them (False).
  default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
  the results object for the pipeline task is returned.
Description
The hsd_restoredata task restores flagged and calibrated data from archived
ASDMs and pipeline flagging and calibration data products. Pending archive
retrieval support hsd_restoredata assumes that the required products
are available in the rawdata_dir in the format produced by the
hifa_exportdata task.
hsd_restoredata assumes that the following entities are available in the raw
data directory
o the ASDMs to be restored
o for each ASDM in the input list
  o a compressed tar file of the final flagversions file, e.g.
    uid___A002_X30a93d_X43e.ms.flagversions.tar.gz
  o a text file containing the applycal instructions, e.g.
    uid___A002_X30a93d_X43e.ms.calapply.txt
  o a compressed tar file containing the caltables for the parent session,
    e.g. uid___A001_X74_X29.session_3.caltables.tar.gz
hsd_restoredata performs the following operations
o imports the ASDM(s))
o removes the default MS.flagversions directory created by the filler
o restores the final MS.flagversions directory stored by the pipeline
o restores the final set of pipeline flags to the MS
o restores the final calibration state of the MS

o restores the final calibration tables for each MS
o applies the calibration tables to each MS
Issues
Examples
1. Restore the pipeline results for a single ASDM in a single session
hsd_restoredata (vis=['uid___A002_X30a93d_X43e'], session=['session_1'], ocorr_mode='ao')

Parameter List

name	type	default	description
vis	stringArray	None	List of input visibility data
session	stringArray	None	List of sessions one per visibility file
products_dir	string	../products	The archived pipeline data products directory
copytoraw	bool	True	Copy calibration and flagging tables to raw data directory
rawdata_dir	string	../rawdata	The raw data directory
lazy	bool	False	Use the lazy filler option
bdfflags	bool	True	Set the BDF flags
ocorr_mode	string	ao	Correlation import mode
asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR	List of tables to import asis
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hsd_skycal

Task Description

Calibrate data
The hsd_skycal generates a caltable for sky calibration that
stores reference spectra, which is to be subtracted from on-source

spectra to filter out non-source contribution.
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
calmode -- Calibration mode. Available options are 'auto' (default),
  'ps', 'otf', and 'otfraster'. When 'auto' is set, the task will
  use preset calibration mode that is determined by inspecting data.
  'ps' mode is simple position switching using explicit reference
  scans. Other two modes, 'otf' and 'otfraster', will generate
  reference data from scans at the edge of the map. Those modes
  are intended for OTF observation and the former is defined for
  generic scanning pattern such as Lissajous, while the latter is
  specific use for raster scan.
  default: 'auto'
  options: 'auto', 'ps', 'otf', 'otfraster'
fraction -- Sub-parameter for calmode. Edge marking parameter for
  'otf' and 'otfraster' mode. It specifies a number of OFF scans
  as a fraction of total number of data points.
  default: '10%'
  options: String style like '20%', or float value less than 1.0.
        For 'otfraster' mode, you can also specify 'auto'.

noff -- Sub-parameter for calmode. Edge marking parameter for 'otfraster'
  mode. It is used to specify a number of OFF scans near edge directly
  instead to specify it by fractional number by 'fraction'. If it is
  set, the value will come before setting by 'fraction'.
  default: -1 (use setting by 'fraction')
  options: any positive integer value
width -- Sub-parameter for calmode. Edge marking parameter for 'otf'
  mode. It specifies pixel width with respect to a median spatial
  separation between neighboring two data in time. Default will
  be fine in most cases.
  default: 0.5
  options: any float value
elongated -- Sub-parameter for calmode. Edge marking parameter for
  'otf' mode. Please set True only if observed area is elongated
  in one direction.
  default: False
pipelinemode -- The pipeline operating mode. In 'automatic' mode the
  pipeline determines the values of all context defined pipeline inputs
  automatically. In 'interactive' mode the user can set the pipeline
  context defined parameters manually. In 'getinputs' mode the user
  can check the settings of all pipeline parameters without running
  the task.
  default: 'automatic'.
---- pipeline context defined parameter argument which can be set only in
'interactive mode'
infiles -- List of data files. These must be a name of MeasurementSets that

are registered to context via hsd_importdata task.
default: []
example: vis=['X227.ms', 'X228.ms']
field -- Data selection by field name.
default: '' (all fields)
spw -- Data selection by spw.
default: '' (all spws)
example: '3,4' (generate caltable for spw 3 and 4)
['0','2'] (spw 0 for first data, 2 for second)
scan -- Data selection by scan number.
default: '' (all scans)
example: '22,23' (use scan 22 and 23 for calibration)
['22','24'] (scan 22 for first data, 24 for second)
pol -- Data selection by polarizations.
default: '' (all polarizations)
example: '0' (generate caltable for pol 0)
['0~1','0'] (pol 0 and 1 for first data, only 0 for second)
--- pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Generate caltables for all data managed by context.
default(hsd_skycal)
hsd_skycal()

Parameter List

name	type	default	description
calmode	string	auto	Calibration mode (default auto)
fraction	any	10%	fraction of the OFF data to mark
noff	int	-1	number of the OFF data to mark
width	double	0.5	width of the pixel for edge detection
elongated	bool	False	whether observed area is elongated in one direction or not
pipelinemode	string	automatic	The pipeline operating mode
infiles	stringArray	None	List of input files to be calibrated (default all)
field	string	None	Field to be calibrated (default all)
spw	any	None	select data by spw ids, e.g. \'3,5,7\' (\'\'=all)
scan	any	None	select data by scan numbers, e.g. \'21~23\' (\'\'=all)
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hsd_tsysflag

Task Description

Flag deviant system temperature measurements
Flag deviant system temperatures for single dish measurements.
Flag all deviant system temperature measurements in the system temperature
calibration table by running a sequence of flagging tests, each designed
to look for a different type of error.
If a file with manual Tsys flags is provided with the 'filetemplate'
parameter, then these flags are applied prior to the evaluation of the
flagging heuristics listed below.
The tests are:
1. Flag Tsys spectra with high median values
2. Flag Tsys spectra with high median derivatives. This is meant to spot
spectra that are 'ringing'.
3. Flag the edge channels of the Tsys spectra in each SpW.
4. Flag Tsys spectra whose shape is different from that associated with
the BANDPASS intent.
5. Flag 'birdies'.
6. Flag the Tsys spectra of all antennas in a timestamp and spw if
proportion of antennas already flagged in this timestamp and spw exceeds

a threshold, and flag Tsys spectra for all antennas and all timestamps
in a spw, if proportion of antennas that are already entirely flagged
in all timestamps exceeds a threshold.
Keyword arguments
--- Pipeline parameter arguments which can be set in any pipeline mode
pipelinemode -- The pipeline operating mode. In 'automatic' mode the pipeline
    determines the values of all context defined pipeline inputs automatically.
    In interactive mode the user can set the pipeline context defined
    parameters manually. In 'getinputs' mode the user can check the settings
    of all pipeline parameters without running the task.
    default: 'automatic'.
flag_nmedian -- True to flag Tsys spectra with high median value.
    default: True
fnm_limit -- Flag spectra with median value higher than fnm_limit * median
    of this measure over all spectra.
    default: 2.0
fnm_byfield -- Evaluate the nmedian metric separately for each field.
    default: True

flag_derivative -- True to flag Tsys spectra with high median derivative.
    default: True
fd_max_limit -- Flag spectra with median derivative higher than
    fd_max_limit * median of this measure over all spectra.
    default: 5.0
flag_edgechans -- True to flag edges of Tsys spectra.
    default: True
fe_edge_limit -- Flag channels whose channel to channel difference >
    fe_edge_limit * median across spectrum.
    default: 3.0
flag_fieldshape -- True to flag Tsys spectra with a radically different
    shape to those of the ff_refintent.
    default: True
ff_refintent -- Data intent that provides the reference shape for
    'flag_fieldshape'.
    default: BANDPASS
ff_max_limit -- Flag Tsys spectra with 'fieldshape' metric values >
    ff_max_limit.
    default: 5.0
flag_birdies -- True to flag channels covering sharp spectral features.
    default: True
fb_sharps_limit -- Flag channels bracketing a channel to channel
    difference > fb_sharps_limit.
    default: 0.05
flag_toomany -- True to flag Tsys spectra for which a proportion of
    antennas for given timestamp and/or proportion of antennas that are
    entirely flagged in all timestamps exceeds their respective thresholds.
    default: True

tmf1_limit -- Flag Tsys spectra for all antennas in a timestamp and spw if
proportion of antennas already flagged in this timestamp and spw exceeds
tmf1_limit.
default: 0.666
tmef1_limit -- Flag Tsys spectra for all antennas and all timestamps
in a spw, if proportion of antennas that are already entirely flagged
in all timestamps exceeds tmef1_limit.
default: 0.666
metric_order -- Order in which to evaluate the flagging metrics that are
enables. Disabled metrics are skipped.
default: 'nmedian,derivative,edgechans,fieldshape,birdies,toomany'
normalize_tsys -- True to create a normalized Tsys table that is used to
evaluate the Tsys flagging metrics. All newly found flags are also applied
to the original Tsys caltable that continues to be used for subsequent
calibration.
default: False
filetemplate -- The name of a text file that contains the manual Tsys flagging
template. If the template flags file is undefined, a name of the form
'msname.flagtsystemplate.txt' is assumed.
default: ''
--- Pipeline context defined parameter arguments which can be set only in
'interactive mode'
caltable -- List of input Tsys calibration tables
default: [] - Use the table currently stored in the pipeline context.
example: caltable=['X132.ms.tsys.s2.tbl']
--- Pipeline task execution modes
dryrun -- Run the commands (True) or generate the commands to be run but
do not execute (False).
default: True
acceptresults -- Add the results of the task to the pipeline context (True) or
reject them (False).
default: True
Output
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
Examples
1. Flag Tsys measurements using currently recommended tests:
hsd_tsysflag()
2. Flag Tsys measurements using all recommended tests apart from that
using the 'fieldshape' metric:

hsd_tsysflag(flag_fieldshape=False)

Parameter List

name	type	default	description
vis	stringArray	None	List of input MeasurementSets (Not used)
caltable	stringArray	None	List of input caltables
flag_nmedian	bool	True	True to flag Tsys spectra with high median value
fnm_limit	double	2.0	Flag spectra with median greater than fnm_limit * median over all spectra
fnm_byfield	bool	True	Evaluate the nmedian metric separately for each field.
flag_derivative	bool	True	True to flag Tsys spectra with high median derivative
fd_max_limit	double	5.0	Flag spectra with median derivative higher than fd_max_limit * median of this measure over all spectra
flag_edgechans	bool	True	True to flag edges of Tsys spectra
fe_edge_limit	double	3.0	Flag channels whose channel to channel difference greater than fe_edge_limit * median across spectrum

flag_fieldshape	bool	True	True to flag Tsys spectra with a radically different shape to those of the ff_refintent
ff_refintent	string	BANDPASS	Data intent providing the reference shape for \'flag_fieldshape\'
ff_max_limit	double	5.0	Flag Tsys spectra with \'fieldshape\' metric greater than ff_max_limit
flag_birdies	bool	True	True to flag channels covering sharp spectral features
fb_sharps_limit	double	0.05	Flag channels bracketing a channel to channel difference greater than fb_sharps_limit
flag_toomany	bool	True	True to flag Tsys spectra for which a proportion of timestamps or proportion of antennas that are entirely flagged exceeds their respective thresholds.
tmf1_limit	double	0.666	Flag all Tsys spectra within a timestamp for an antenna if proportion flagged already exceeds tmf1_limit

tmef1_limit	double	0.666	Flag all Tsys spectra for all antennas in a spw, if proportion of antennas that are already entirely flagged in all timestamps exceeds tmef1_limit
metric_order	string	nmedian,derivative,edgechans,fieldshape,birdies,toomany	Order in which to evaluate the flagging metric(s); inactive metrics are skipped.
normalize_tsys	bool	False	Normalize Tsys prior to computing the flagging metric(s)
filetemplate	stringArray	None	File containing manual Tsys flags to apply.
pipelinemode	string	automatic	The pipeline operations mode
dryrun	bool	False	Run the task (False) or list commands (True)
acceptresults	bool	True	Automatically apply results to context

Back

Created at Fri Sep 14 10:04:26 2018 UTC
This page is automatically generated from XML interface definition files.