List of Heuristics Tasks (Pipeline 2021.2.0.128)

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 flags and calibration state from a pipeline run
h_resume	Restore a save pipeline state
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

23 tasks available.

task name	description
hif_analyzealpha	Base analyzealpha task
hif_antpos	Derive an antenna position calibration table
hif_applycal	Apply the calibration(s) to the data
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_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	Create new MeasurementSets for science target 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_transformimagedata	Base transformimagedata task
hif_uvcontfit	Fit the continuum in the UV plane
hif_uvcontsub	Subtract the fitted continuum from the data

Interferometry ALMA

26 tasks available.

task name	description

hifa_antpos	Derive an antenna position calibration table
hifa_bandpassflag	Bandpass calibration 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 meta data based flagging of a list of MeasurementSets.
hifa_flagtargets	Do science target flagging
hifa_fluxcalflag	Locate and flag line regions in solar system flux calibrators
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	Calculates the best robust value and Briggs weighting parameter to achieve sensitivity and angular resolution goals.
hifa_importdata	Imports data into the interferometry pipeline
hifa_lock_refant	Lock reference antenna list
hifa_polcalflag	Flag polarization calibrators
hifa_renorm	Base renorm task
hifa_restoredata	Restore flagged and calibration interferometry data from a pipeline run
hifa_session_bandpass	Compute bandpass calibration solutions
hifa_session_refant	Select best reference antenna for session(s)
hifa_spwphaseup	Compute phase calibration spw map and per spw phase offsets
hifa_targetflag	Flag target source outliers
hifa_timegaincal	Determine temporal gains from calibrator observations
hifa_tsysflag	Flag deviant system temperatures for ALMA interferometry measurements. This is done by running a sequence of flagging subtasks, each looking for a different type of possible error.
hifa_unlock_refant	Unlock reference antenna list
hifa_wvrgcalflag	Generate a gain table based on Water Vapor Radiometer data, interpolating over antennas with bad radiometers.

hifa_wvrgcal	Generate a gain table based on Water Vapor Radiometer data, and calculate a QA score based on its effect on the interferometric data.

Interferometry ALMA SRDP

1 tasks available.

task name	description
hifas_imageprecheck	Calculates the best robust value and Briggs weighting parameter to achieve sensitivity and angular resolution goals.

Interferometry VLA

31 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_fixpointing	Base fixpointing task
hifv_flagbaddef	Flagging of deformatters - amp and phase
hifv_flagcal	Flagcal task
hifv_flagdata	Do basic deterministic flagging of a list of MeasurementSets
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, tec_maps, switched power.
hifv_restoredata	Restore flagged and calibration interferometry data from a pipeline run
hifv_rqcal	Runs gencal in rq mode
hifv_selfcal	Selfcal task executing gaincal and applycal
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
hifv_vlassmasking	Base vlassmasking task

Single-Dish

12 tasks available.

task name	description
hsd_applycal	Apply the calibration(s) to the data
hsd_atmcor	Apply offline ATM correction 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 Wed Sep 8 10:27:27 2021 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
1. Apply the calibration to the target data
hif_applycal (intent='TARGET')

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets in the pipeline context. Parameter is not available when pipelinemode='automatic'. example: ['X227.ms']

field	string	None	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. Parameter is not available when pipelinemode='automatic'. example: '3C279', '3C279, M82'
intent	string	None	A string containing the list of intents against which the selected fields will be matched. Defaults to all supported intents in the pipeline context. Parameter is not available when pipelinemode='automatic'. example: '*TARGET*'
spw	string	None	The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context. Parameter is not available when pipelinemode='automatic'. example: '17', '11, 15'
antenna	string	None	The list of antennas to which the calibration will be applied. Defaults to all antennas. Not currently supported. Parameter is not available when pipelinemode='automatic'.
applymode	string	None	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)
flagbackup	bool	True	Backup the flags before the apply
flagsum	bool	True	Compute before and after flagging summary statistics
flagdetailedsum	bool	False	Compute detailed before and after flagging statistics summaries. Parameter available only when if flagsum is True.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run task (False) or display the command(True). Parameter is available only when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Parameter is available only when pipelinemode='interactive'.

Back

---

h_export_calstate

Task Description

Save the pipeline calibration state to disk

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

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	stringVec	None	List of visibility data files for which flagging and calibration information will be exported. Defaults to the list maintained in the pipeline context. Can only be set in pipelinemode='interactive' example: vis=['X227.ms', 'X228.ms']
session	stringVec	None	session -- List of sessions one per visibility file. Defaults to a single virtual session containing all the visibility files in vis. Can only be set in pipelinemode='interactive' example: session=['session1', 'session2']
imaging_products_only	bool	False	Export the science target image products only
exportmses	bool	False	Export MeasurementSets defined in vis instead of flags, caltables, and calibration instructions. Can only be set in pipelinemode='interactive' example: exportmses = True
pprfile	string	None	Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'. Can only be set in pipelinemode='interactive' example: pprfile=['PPR_GRB021004.xml']
calintents	string	None	calintents -- List of calibrator image types to be exported. Defaults to all standard calibrator intents 'BANDPASS', 'PHASE', 'FLUX' Can only be set in pipelinemode='interactive' example: calintents='PHASE'
calimages	stringVec	None	List of calibrator images to be exported. Defaults to all calibrator images recorded in the pipeline context. Can only be set in pipelinemode='interactive' example: calimages=['3C454.3.bandpass', '3C279.phase']
targetimages	stringVec	None	List of science target images to be exported. Science target images recorded in the pipeline context. Can only be set in pipelinemode='interactive' example: targetimages=['NGC3256.band3', 'NGC3256.band6']
products_dir	string	None	Name of the data products subdirectory. Can only be set in pipelinemode='interactive' example: products_dir='../products'

pipelinemode	string	automatic	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.
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

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

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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
session	stringVec	None	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. example: session=['session_1', 'session_2']
pipelinemode	string	automatic	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.
asis	string	None	ASDM tables to convert as is default: 'Antenna Station Receiver CalAtmosphere' Can only be set in pipelinemode='interactive' example: 'Receiver', ''
process_caldevice	bool	False	Ingest the ASDM caldevice table. Can only be set in pipelinemode='interactive'
overwrite	bool	False	Overwrite existing files on import. Can only be set in pipelinemode='interactive'. When converting ASDM to MS, if overwrite=False and the MS already exists in output directory, then this existing MS dataset will be used instead.
nocopy	bool	False	When importing an MS, disable copying of the MS to the working directory. Can only be set in pipelinemode='interactive'.
bdfflags	bool	True	Apply BDF flags on import. Can only be set in pipelinemode='interactive'.

lazy	bool	False	Use the lazy import option. Can only be set in pipelinemode='interactive'.
ocorr_mode	string	ca	Read in cross- and auto-correlation data(ca), cross-correlation data only (co), or autocorrelation data only (ao).
createmms	string	automatic	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. Can only be set in pipelinemode='interactive'
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
Examples
1. Create the pipeline context
h_init()

Parameter List

name	type	default	description
pipelinemode	string	automatic	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.
loglevel	string	info	Log level for pipeline messages. Log messages below this threshold will not be displayed.
plotlevel	string	default	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.
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

Examples

1. Create a 4X channel smoothed output MS from the input MS
   h_mssplit(chanbin=4)

Parameter List

name	type	default	description
vis	stringVec	None	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. Can only use with pipelinemode='interactive' example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
outputvis	stringVec	None	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 Can only use with pipelinemode='interactive' example: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
field	string	None	Set of data selection field names or ids, \'\' for all
intent	string	None	Select intents to split default: '', All data is selected. Can only use with pipelinemode='interactive' example: 'TARGET'
spw	string	None	Select spectral windows to split. Can only use with pipelinemode='interactive' default: '', All spws are selected example: '9', '9,13,15'
datacolumn	string	data	Select spectral windows to split. The standard CASA options are supported. Can only use with pipelinemode='interactive' example: 'corrected', 'model'
chanbin	int	1	The channel binning factor. 1 for no binning, otherwise 2, 4, 8, or 16. Can only use with pipelinemode='interactive' example: 2, 4
timebin	string	0s	The time binning factor. '0s' for no binning Can only use with pipelinemode='interactive' example: '10s' for 10 second binning
replace	bool	True	If a split was performed delete the parent MS and remove it from the context. Can only use with pipelinemode='interactive'
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display the command(True)
acceptresults	bool	True	Add the results to the pipeline context

h_restoredata

Task Description

Restore flags and calibration state from a pipeline run

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	stringVec	None	List of raw visibility data files to be restored. Assumed to be in the directory specified by rawdata_dir. example: vis=['uid___A002_X30a93d_X43e'] Can only set when pipelinemode='interactive'
session	stringVec	None	List of sessions, one per visibility file. example: session=['session_3'] Can only set when pipelinemode='interactive'
products_dir	string	../products	Path to the data products directory, used to copy calibration products from. The parameter is effective only when copytoraw=True. When copytoraw=False, calibration products in rawdata_dir will be used. example: products_dir='/path/to/my/products' Can only set when pipelinemode='interactive'
copytoraw	bool	True	Copy calibration and flagging tables from products_dir to rawdata_dir directory. example: copytoraw=False Can only set when pipelinemode='interactive'
rawdata_dir	string	../rawdata	Path to the rawdata subdirectory. example: rawdata_dir='/path/to/my/rawdata' Can only set when pipelinemode='interactive'
lazy	bool	False	Use the lazy filler option example: lazy=True Can only set when pipelinemode='interactive'
bdfflags	bool	True	Set the BDF flags example: bdfflags=False Can only set when pipelinemode='interactive'
ocorr_mode	string	ca	Set correlation import mode example: ocorr_mode='ca' Can only set when pipelinemode='interactive'

asis	string	None	Set list of tables to import as-is into the Measurement Set example: ocorr_mode='Source Receiver' Can only set when pipelinemode='interactive'
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). Can only set when pipelinemode='interactive'
acceptresults	bool	True	Add the results into the pipeline context

Back

---

h_resume

Task Description

Restore a save pipeline state

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	Name of the saved pipeline state. Setting filename to 'last' restores the most recently saved pipeline state whose name begins with 'context*'.

Back

---

h_save

Task Description

Save the pipeline state to disk

1. Save the current state in the default file
   h_save()
2. Save the current state to a file called 'savestate_1'
   h_save(filename='savestate_1')

Parameter List

name	type	default	description
filename	string	None	Name of the saved pipeline state. If filename is '' then a unique name will be generated computed several components: the root, 'context', the current stage number, and the time stamp.

Back

---

h_show_calstate

Task Description

Show the current pipeline calibration state

Parameter List

No parameter

Back

---

h_tsyscal

Task Description

Derive a Tsys calibration table

Parameter List

name	type	default	description
vis	stringVec	None	List of input visibility files. Parameter is not available when pipelinemode='automatic'. example: vis=['ngc5921.ms']
caltable	stringVec	None	Name of output gain calibration tables. Parameter is not available when pipelinemode='automatic'. example: caltable='ngc5921.gcal'
chantol	int	1	The tolerance in channels for mapping atmospheric calibration windows (TDM) to science windows (FDM or TDM). example: chantol=5
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or list commands (True). Parameter is available only when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or Parameter is available only when pipelinemode='interactive'.

Back

---

h_weblog

Task Description

Open the pipeline weblog in a browser

Parameter List

name	type	default	description
pipelinemode	string	automatic	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.
relpath	string	None	Relative path to the weblog index file. This file must be located in a child directory of the CASA working directory. If relpath is left unspecified, the most recent weblog will be located and displayed.

Back

Created at Wed Sep 8 10:27:27 2021 UTC
This page is automatically generated from XML interface definition files.

Summary of generic interferometric tasks and parameters

hif_analyzealpha

Task Description

Base analyzealpha task
Examples
1. Basic analyzealpha task
hif_analyzealpha()

Parameter List

name	type	default	description
vis	stringVec	None	List of visisbility 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. example: vis=['X227.ms', 'asdms.tar.gz']
image	string	None	Restored subimage
alphafile	string	None	Input spectral index map
alphaerrorfile	string	None	Input spectral index error map
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

hif_antpos

Task Description

Derive an antenna position calibration table

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	stringVec	None	List of input visibility files. Not available when pipelinemode='automatic'. example: ['ngc5921.ms']
caltable	stringVec	None	Name of output gain calibration tables. Not available when pipelinemode='automatic'. example: ['ngc5921.gcal']
hm_antpos	string	manual	Heuristics method for retrieving the antenna position corrections. The options are 'online' (not yet implemented), 'manual', and 'file'.
antenna	string	None	The list of antennas for which the positions are to be corrected. Available when hm_antpos='manual'. example: antenna='DV05,DV07'
offsets	doubleVec	None	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. Available when hm_antpos='manual'. example: offsets=[0.01, 0.02, 0.03, 0.03, 0.02, 0.01]
antposfile	string	None	The file(s) containing the antenna offsets. Used if hm_antpos is 'file'. example: 'antennapos.csv'
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or list commands (True). Available when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Available when pipelinemode='interactive'.

Back

---

hif_applycal

Task Description

Apply the calibration(s) to the data

1. Apply the calibration to the target data

hif_applycal (intent='TARGET')

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets in the pipeline context. Parameter is not available when pipelinemode='automatic'. example: ['X227.ms']
field	string	None	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. Parameter is not available when pipelinemode='automatic'. example: '3C279', '3C279, M82'
intent	string	None	A string containing the list of intents against which the selected fields will be matched. Defaults to all supported intents in the pipeline context. Parameter is not available when pipelinemode='automatic'. example: '*TARGET*'
spw	string	None	The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context. Parameter is not available when pipelinemode='automatic'. example: '17', '11, 15'
antenna	string	None	The list of antennas to which the calibration will be applied. Defaults to all antennas. Not currently supported. Parameter is not available when pipelinemode='automatic'.
applymode	string	None	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)
calwt	boolVec	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 before and after flagging statistics summaries. Parameter available only when if flagsum is True.

pipelinemode	string	automatic	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.
dryrun	bool	False	Run task (False) or display the command(True). Parameter is available only when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Parameter is available only when pipelinemode='interactive'.

Back

---

hif_bandpass

Task Description

Compute bandpass calibration solutions
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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
caltable	stringVec	None	The list of output calibration tables. Defaults to the standard pipeline naming convention. Example: caltable=['M82.gcal', 'M82B.gcal']
field	string	None	The list of field names or field ids for which bandpasses are computed. Defaults to all fields. Examples: field='3C279', field='3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to all data with bandpass intent. Example: intent='*PHASE*'

spw	string	None	The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows. Example: spw='11,13,15,17'
antenna	string	None	Set of data selection antenna IDs
phaseup	bool	True	Do a phaseup on the data before computing the bandpass solution.
phaseupsolint	any	int	The phase correction solution interval in CASA syntax. Used when phaseup is True. Example: phaseupsolint=300
phaseupbw	string	None	Bandwidth to be used for phaseup. Defaults to 500MHz. Used when phaseup is True. Examples: phaseupbw='' to use entire bandpass phaseupbw='500MHz' to use central 500MHz
solint	any	inf	Time and channel solution intervals in CASA syntax. Examples: solint='inf,10ch', 'inf'
combine	string	scan	Data axes to combine for solving. Axes are '', 'scan', 'spw', 'field' or any comma-separated combination. Example: combine='scan,field'
refant	string	None	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. Examples: refant='DV01', refant='DV06,DV07'
solnorm	bool	True	Normalise the bandpass solution
minblperant	int	4	Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions.
minsnr	double	3.0	Reject solutions below this SNR
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

hif_checkproductsize

Task Description

Check imaging product size

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. \'\': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
maxcubesize	double	-1.0	Maximum allowed cube size in gigabytes (mitigation goal) -1: automatic from performance parameters
maxcubelimit	double	-1.0	Maximum allowed cube limit in gigabytes (mitigation failure limit) -1: automatic from performance parameters
maxproductsize	double	-1.0	Maximum allowed product size in gigabytes (mitigation goal and failure limit) -1: automatic from performance parameters
maximsize	double	-1.0	Maximum allowed image count size (mitigation goal and hard maximum). Parameter maximsize must be even and divisible by 2,3,5,7 only. Note that maximsize is disabled by default and cannot be set at the same time as maxcubesize, maxcubelimit and maxproductsize! -1: disables mitigation for this parameter
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
parallel	string	automatic	Use MPI cluster where possible
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

hif_correctedampflag

Task Description

Flag corrected - model amplitudes based on calibrators.
Run default flagging on bandpass calibrator with recommended settings:
hif_correctedampflag()

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
intent	string	None	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. Example: intent='*PHASE*'
field	string	None	The list of field names or field ids for which bandpasses are computed. If undefined (default), it will select all fields. Examples: field='3C279', '3C279, M82'
spw	string	None	The list of spectral windows and channels for which bandpasses are computed. If undefined (default), it will select all science spectral windows. Example: spw='11,13,15,17'
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. If an iteration results in no new flags, then subsequent iterations are skipped.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

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	stringVec	None	Image X and Y cell size(s) with units or pixels per beam. Single value same for both. \'ppb\' for pixels per beam. 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'
cfcache	string	None	Convolution function cache directory name

conjbeams	bool	False	Use conjugate frequency in tclean for wideband A-terms.
cyclefactor	double	-999.0	Controls the depth of clean in minor cycles based on PSF.
cycleniter	int	-999	Controls max number of minor cycle iterations in a single major cycle.
datacolumn	string	None	Data column to image
deconvolver	string	None	Minor cycle algorithm (multiscale or mtmfs)
editmode	string	None	The edit mode of the task (e.g. add, edit). Defaults to 'add'.
field	stringVec	None	Set of data selection field names or ids.
imaging_mode	string	None	Identity of product type (e.g. VLASS quick look) desired. This will determine the heuristics used.
imsize	any	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.
pbmask	float	None	Used to declare primary beam gain level for cleaning with primary beam mask (usemask='pb'), used only for VLASS-SE-CONT imaging mode.
nbin	int	-1	Channel binning factor.
nchan	int	-1	Number of channels, -1 = all
niter	int	0	The max total number of minor cycle iterations allowed for tclean
nterms	int	0	Number of Taylor coefficients in the spectral model
parameter_file	string	None	keyword=value text file as alternative method of input parameters
pblimit	double	-999.0	PB gain level at which to cut off normalizations
phasecenter	any	None	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'

reffreq	string	None	Reference frequency of the output image coordinate system
restfreq	string	None	List of rest frequencies or a rest frequency in a string for output image.
robust	double	-999.0	Briggs robustness parameter for tclean
scales	any	None	The scales for multi-scale imaging.
specmode	string	None	Spectral gridding type (mfs, cont, cube, \'\' for default)
spw	any	None	Set of data selection spectral window/channels, \'\' for all
start	any	None	First channel for frequency mode images. Starts at first input channel of the spw. example: '22.3GHz'
stokes	string	None	Stokes Planes to make
sensitivity	double	0.0	None
threshold	string	None	Stopping threshold (number in units of Jy, or string)
nsigma	double	-999.0	Multiplicative factor for rms-based threshold stopping
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
pipelinemode	string	automatic	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.
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

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. \'\': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
target_list	any	{}	Dictionary specifying targets to be imaged; blank will read list from context
hm_mosweight	any	None	Mosaic weighting Defaults to '' to enable the automatic heuristics calculation. Can be set to True or False manually.
hm_perchanweightdensity	any	None	Calculate the weight density for each channel independently Defaults to '' to enable the automatic heuristics calculation. Can be set to True or False manually.
hm_weighting	any	None	Weighting scheme (natural,uniform,briggs,briggsabs[experimental],briggsbwtaper[experimental])
parallel	string	automatic	Use MPI cluster where possible
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hif_gaincal

Task Description

Determine temporal gains from calibrator observations
Compute standard per scan gain solutions that will be used to calibrate
the target:
hif_gaincal()

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']

caltable	stringVec	None	The list of output calibration tables. Defaults to the standard pipeline naming convention. Example: caltable=['M82.gcal', 'M82B.gcal']
field	string	None	The list of field names or field ids for which gain solutions are to be computed. Defaults to all fields with the standard intent. Example: field='3C279', field='3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to *PHASE*. Examples: intent='', intent='*AMP*,*PHASE*'
spw	string	None	The list of spectral windows and channels for which gain solutions are computed. Defaults to all science spectral windows. Examples: spw='3C279', spw='3C279, M82'
antenna	string	None	Set of data selection antenna ids
hm_gaintype	string	gtype	The type of gain calibration. The options are 'gtype' and 'gspline' for CASA gain types = 'G' and 'GSPLINE' respectively.
calmode	string	ap	Type of solution. The options are 'ap' (amp and phase), 'p' (phase only) and 'a' (amp only). Examples: calmode='p', calmode='a', calmode='ap'
solint	any	inf	Time solution intervals in CASA syntax. Works for hm_gaintype='gtype' only. Examples: solint='inf', solint='int', solint='100sec'
combine	string	None	Data axes to combine for solving. Options are '', 'scan', 'spw', 'field' or any comma-separated combination. Works for hm_gaintype='gtype' only.
refant	string	None	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. Examples: refant='DV01', refant='DV05,DV07'
refantmode	string	None	Controls how the refant is applied. Currently available choices are 'flex', 'strict', and the default value of ''. Setting to '' allows the pipeline to select the appropriate mode based on the state of the reference antenna list. Examples: refantmode='strict', refantmode=''
solnorm	bool	False	Normalize average solution amplitudes to 1.0
minblperant	int	4	Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions. Works for hm_gaintype='gtype' only.

minsnr	double	3.0	Solutions below this SNR are rejected. Works for hm_gaintype='channel' only.
smodel	doubleVec	None	Point source Stokes parameters for source model (experimental). Defaults to using standard MODEL_DATA column data. Example: smodel=[1,0,0,0] - (I=1, unpolarized)
splinetime	double	3600.0	Spline timescale (sec). Used for hm_gaintype='gspline'. Typical splinetime should cover about 3 to 5 calibrator scans.
npointaver	int	3	Tune phase-unwrapping algorithm. Used for hm_gaintype='gspline'. Keep at default value.
phasewrap	double	180.0	Wrap the phase for changes larger than this amount (degrees). Used for hm_gaintype='gspline'. Keep at default value.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hif_lowgainflag

Task Description

Flag antennas with low or high gain

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
intent	string	None	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.
spw	string	None	The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context. Examples: spw='17', spw='11, 15'
refant	string	None	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. Examples: refant='DV01', refant='DV06,DV07'
flag_nmedian	bool	True	True to flag figures of merit greater than fnm_hi_limit * median or lower than fnm_lo_limit * median.
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 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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hif_makecutoutimages

Task Description

Base makecutoutimages task

1. Basic makecutoutimages task

   hif_makecutoutimages()

Parameter List

name	type	default	description
vis	stringVec	None	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. (can be set only in 'interactive mode') example: vis=['X227.ms', 'asdms.tar.gz']
offsetblc	doubleVec	None	-x and -y offsets to the bottom lower corner (blc) in arcseconds (can be set in any pipeline mode)
offsettrc	doubleVec	None	+x and +y offsets to the top right corner (trc) in arcseconds (can be set in any pipeline mode)
pipelinemode	string	automatic	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. (can be set in any pipeline mode)
dryrun	bool	False	Run the commands (False) or generate the commands to be run but do not execute (True).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hif_makeimages

Task Description

Compute clean map

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
target_list	any	{}	Dictionary specifying targets to be imaged; blank will read list from context
hm_masking	string	auto	Clean masking mode. Options are 'centralregion', 'auto', 'manual' and 'none'
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	any	None	Do pruning on the grow mask Defaults to '' to enable the automatic heuristics calculation. Can be set to True or False manually.
hm_minpercentchange	double	-999.0	Mask size change threshold
hm_fastnoise	any	None	Faster noise calucation for automask or nsigma stopping Defaults to '' to enable the automatic heuristics calculation. Can be set to True or False manually.
hm_nsigma	double	0.0	Multiplicative factor for rms-based threshold stopping
hm_perchanweightdensity	any	None	Calculate the weight density for each channel independently Defaults to '' to enable the automatic heuristics calculation. Can be set to True or False manually.
hm_npixels	int	0	Number of pixels to determine uv-cell size for super-uniform weighting
hm_cyclefactor	double	-999.0	Scaling on PSF sidelobe level to compute the minor-cycle stopping threshold
hm_minpsffraction	double	-999.0	PSF fraction that marks the max depth of cleaning in the minor cycle
hm_maxpsffraction	double	-999.0	PSF fraction that marks the minimum depth of cleaning in the minor cycle
hm_weighting	any	None	Weighting scheme (natural,uniform,briggs,briggsabs[experimental],briggsbwtaper[experimental])
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
cleancontranges	bool	False	Clean continuum frequency ranges in cubes
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
hm_mosweight	any	None	Mosaic weighting Defaults to '' to enable the automatic heuristics calculation. Can be set to True or False manually.
overwrite_on_export	bool	True	Replace existing image products when h/hifa/hifv_exportdata is called. If False, images that would have the same FITS name on export, are amended to include a version number. For example, if oussid.J1248-4559_ph.spw21.mfs.I.pbcor.fits would already be exported by a previous call to hif_makeimags, then 'oussid.J1248-4559_ph.spw21.mfs.I.pbcor.v2.fits' would also be exported to the products/ directory. The first exported product retains the same name. Additional products start counting with 'v2', 'v3', etc.
parallel	string	automatic	Clean images using MPI cluster
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just 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
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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. "": use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
imagename	string	None	Prefix for output image names, "" for automatic.
intent	string	TARGET	Select intents for which associated fields will be imaged. Possible choices are PHASE, BANDPASS, AMPLITUDE, CHECK and TARGET or combinations thereof. Examples: 'PHASE,BANDPASS', 'TARGET'
field	string	None	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". "" Fields matching intent, one image per target source.
spw	string	None	Select spectral windows to image. "": Images will be computed for all science spectral windows.
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	Select a set of uv ranges to image. "": All uv data is included Examples: '0~1000klambda', ['0~100klambda', 100~1000klambda]
specmode	string	None	Frequency imaging mode, 'mfs', 'cont', 'cube', 'repBW'. '' defaults to 'cube' if intent parameter includes 'TARGET' otherwise 'mfs'. 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
outframe	string	None	velocity frame of output image (LSRK, '' for automatic)
hm_imsize	any	None	Image X and Y size in pixels or PB level for single fields. The explicit sizes must be even and divisible by 2,3,5,7 only. 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. Examples: '0.3pb', [120, 120]
hm_cell	stringVec	None	Image X and Y cell sizes. "" computes the cell size based on the UV coverage of all the fields to be imaged and uses a 5 pix per beam sampling. The pix per beam specification ('ppb') uses the above default cell size ('5ppb') and scales it accordingly. The cells can also be specified as explicit measures. Examples: '3ppb', ['0.5arcsec', '0.5arcsec']
calmaxpix	int	300	Maximum image X or Y size in pixels if a calibrator is being imaged ('PHASE', 'BANDPASS', 'AMPLITUDE' or 'FLUX' intent).
phasecenter	any	None	Direction measure or field id of the image center. The default phase center is set to the mean of the field directions of all fields that are to be image together. Examples: 'J2000 19h30m00 -40d00m00', 0
nchan	int	-1	Total number of channels in the output image(s) -1 selects enough channels to cover the data selected by spw consistent with start and width.
start	any	None	Start of image frequency axis as frequency or velocity. "" selects start frequency automatically.

width	any	None	Output channel width. Difference in frequency between 2 selected channels for frequency mode images. 'pilotimage' for 15 MHz / 8 channel heuristic
nbins	any	None	Channel binning factors for each spw. Format: 'spw1:nb1,spw2:nb2,...' with optional wildcards: '*:nb' Examples: '9:2,11:4,13:2,15:8', '*:2'
robust	double	-999.0	Briggs robustness parameter Values range from -2.0 (uniform) to 2.0 (natural)
uvtaper	stringVec	None	uv-taper on outer baselines
clearlist	bool	True	Clear any existing target list
per_eb	bool	False	Make an image target per EB
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
parallel	string	automatic	Use MPI cluster where possible
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hif_makermsimages

Task Description

Base makermsimages task

1. Basic makermsimages task
   hif_makermsimages()

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hif_mstransform

Task Description

Create new MeasurementSets for science target imaging

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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
outputvis	stringVec	None	The list of output transformed MeasurementSets to be used for imaging. The output list must be the same length as the input list. The default output name defaults to _target.ms Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
field	string	None	Select fields name(s) or id(s) to transform. Only fields with data matching the intent will be selected. Examples: '3C279', 'Centaurus*', '3C279,J1427-421'
intent	string	None	Select intents for which associated fields will be imaged. By default only TARGET data is selected. Examples: 'PHASE,BANDPASS'
spw	string	None	Select spectral window/channels to image. By default all science spws for which the specified intent is valid are selected .
chanbin	int	1	Width (bin) of input channels to average to form an output channel. If chanbin > 1 then chanaverage is automatically switched to True.
timebin	string	0s	Bin width for time averaging. If timebin > 0s then timeaverage is automatically switched to True.
pipelinemode	string	automatic	The pipeline operating mode
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

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	stringVec	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_rawflagchans

Task Description

Flag deviant baseline/channels in raw data
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	stringVec	None	List of input MeasurementSets. default: [] - Use the MeasurementSets currently known to the pipeline context.
spw	string	None	The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context. example: spw='17', spw='11, 15'
intent	string	None	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. example: intent='*BANDPASS*'
flag_hilo	bool	True	True to flag channel/baseline data further from the view median than fhl_limit * MAD.
fhl_limit	double	20.0	If flag_hilo is True then flag channel/baseline data further from the view median than fhl_limit * MAD.
fhl_minsample	double	5	Do no flagging if the view median and MAD are derived from fewer than fhl_minsample view pixels.
flag_bad_quadrant	bool	True	True to search for and flag bad antenna quadrants and baseline quadrants. Here a /'quadrant/' is one quarter of the channel axis.
fbq_hilo_limit	double	8.0	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.
fbq_antenna_frac_limit	double	0.2	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.
fbq_baseline_frac_limit	double	1.0	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.

pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	This parameter has no effect. The Tsyscal file is already in the pipeline context and is flagged in situ.

Back

---

hif_refant

Task Description

Select the best reference antennas

1. Compute the references antennas to be used for bandpass and gain calibration.
   hif_refant()

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets in the pipeline context. Not available when pipelinemode='automatic'. example: ['M31.ms']
field	string	None	The comma delimited list of field names or field ids for which flagging scores are computed if hm_refant='automatic' and flagging = True Not available when pipelinemode='automatic'. example: '' (Default to fields with the specified intents), '3C279', '3C279,M82'
spw	string	None	A string containing the comma delimited list of spectral window ids for which flagging scores are computed if hm_refant='automatic' and flagging = True. Not available when pipelinemode='automatic'. example: '' (all spws observed with the specified intents), '11,13,15,17'
intent	string	AMPLITUDE,BANDPASS,PHASE,POLARIZATION	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to all supported intents. Not available when pipelinemode='automatic'. example: 'BANDPASS', 'AMPLITUDE,BANDPASS,PHASE,POLARIZATION'
hm_refant	string	automatic	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.
refant	string	None	The user supplied reference antenna for hm_refant='manual'. If no antenna list is supplied an empty list is returned. example: 'DV05'
geometry	bool	True	Score antenna by proximity to the center of the array. This option is quick as only the ANTENNA table must be read. Parameter is available when hm_refant='automatic'.
flagging	bool	True	Score antennas by percentage of unflagged data. This option requires computing flagging statistics. Parameter is available when hm_refant='automatic'.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display the command (True) Available when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Available when pipelinemode='interactive'.
refantignore	string	None	string list to be ignored as reference antennas. example: refantignore='ea02,ea03'

Back

hif_setjy

Task Description

Fill the model column with calibrated visibilities

1. Set the model flux densities for all the amplitude calibrators:
   hif_setjy()

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets defined in the pipeline context.
field	string	None	The list of field names or field ids for which the models are to be set. Defaults to all fields with intent '*AMPLITUDE*'. example: field='3C279', field='3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to all data with amplitude intent. example: intent='*AMPLITUDE*'
spw	string	None	The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows. example: spw='11,13,15,17'
model	string	None	Model image for setting model visibilities. Not fully supported. example: see details in help for CASA setjy task
reffile	string	None	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. example: reffile='', reffile='working/flux.csv'
normfluxes	bool	False	Normalize lookup fluxes.
reffreq	string	1GHz	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. example: reffreq='86.0GHz', reffreq='4.65e9Hz'

fluxdensity	any	-1	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. example: [3.06,0.0,0.0,0.0]
spix	double	0.0	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.
scalebychan	bool	True	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.
standard	variant	None	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
pipelinemode	string	automatic	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'.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hif_setmodels

Task Description

Set calibrator source models

1. Set model fluxes for the flux, bandpass, phase, and check sources.
   hif_setmodels()

Parameter List

name	type	default	description
vis	string	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. example: ['M32A.ms', 'M32B.ms']
reference	variant	None	A string containing a comma delimited list of field names defining the reference calibrators. Defaults to field names with intent 'AMPLITUDE'. example: 'M82,3C273'
refintent	string	AMPLITUDE	A string containing a comma delimited list of intents used to select the reference calibrators. Defaults to 'AMPLITUDE'. example: 'BANDPASS'
transfer	variant	None	A string containing a comma delimited list of field names defining the transfer calibrators. Defaults to field names with intent ''. example: 'J1328+041,J1206+30'
transintent	string	BANDPASS	A string containing a comma delimited list of intents defining the transfer calibrators. Defaults to 'BANDPASS,PHASE,CHECK'. '' stands for no transfer sources. example: 'PHASE'
reffile	string	None	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 example: 'myfluxes.csv'
normfluxes	bool	True	
scalebychan	bool	True	Scale the flux density on a per channel basis or else on a per spw basis example: False
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

hif_transformimagedata

Task Description

Base transformimagedata task

1. Basic transformimagedata task
   hif_transformimagedata()

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
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	Select spectral windows to split. The standard CASA options are supported example: 'data', 'model'
chanbin	int	1	
timebin	string	0s	Bin width for time averaging.
replace	bool	False	If a split was performed delete the parent MS and remove it from the context. example: True or False
clear_pointing	bool	True	Clear the pointing table.
modify_weights	bool	False	Re-initialize the weights.
wtmode	string	None	optional weight initialization mode when modify_weights=True
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

hif_uvcontfit

Task Description

Fit the continuum in the UV plane

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
caltable	stringVec	None	The list of output Mueller matrix calibration tables one per input MS. '': The output names default to the standard pipeline name scheme
contfile	string	None	Name of the input file of per source / spw continuum regions '': Defaults first to the file named in the context, next to a file called 'cont.dat' in the pipeline working directory.
field	string	None	The list of field names or field ids for which UV continuum fits are computed. Defaults to all fields. Eexamples: '3C279', '3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. '': Defaults to all data with TARGET intent.
spw	string	None	The list of spectral windows and channels for which uv continuum fits are computed. '', Defaults to all science spectral windows. Example: '11,13,15,17'
combine	string	None	Data axes to be combined for solving. Axes are 'scan', 'spw', or ''. This option is currently not supported.
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. 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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

hif_uvcontsub

Task Description

Subtract the fitted continuum from the data

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
field	string	None	The list of field names or field ids for which UV continuum fits are computed. Defaults to all fields. Eexamples: '3C279', '3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. '': Defaults to all data with TARGET intent.
spw	string	None	The list of spectral windows and channels for which uv continuum fits are computed. '', Defaults to all science spectral windows. Example: '11,13,15,17'
applymode	string	None	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)
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Created at Wed Sep 8 10:27:27 2021 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
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	stringVec	None	List of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. Example: vis=['ngc5921.ms']
caltable	stringVec	None	List of names for the output calibration tables. Defaults to the standard pipeline naming convention. Example: caltable=['ngc5921.gcal']
hm_antpos	string	file	Heuristics method for retrieving the antenna position corrections. The options are 'online' (not yet implemented), 'manual', and 'file'. Example: hm_antpos='manual'
antenna	string	None	The list of antennas for which the positions are to be corrected if hm_antpos is 'manual'. Example: antenna='DV05,DV07'
offsets	doubleVec	None	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. Example: offsets=[0.01, 0.02, 0.03, 0.03, 0.02, 0.01]
antposfile	string	antennapos.csv	The file(s) containing the antenna offsets. Used if hm_antpos is 'file'.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Automatically accept the results of the task into the pipeline context (True) or reject them (False).

Back

---

hifa_bandpassflag

Task Description

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

Parameter List

name	type	default	description
vis	stringVec	None	List of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. Example: vis=['ngc5921.ms']
caltable	stringVec	None	List of names for the output calibration tables. Defaults to the standard pipeline naming convention. Example: caltable=['ngc5921.gcal']
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Set to intent='' by default, which means the task will select all data with the BANDPASS intent. Example: intent='*PHASE*'
field	string	None	The list of field names or field ids for which bandpasses are computed. Set to field='' by default, which means the task will select all fields. Example: field='3C279', field='3C279,M82'
spw	string	None	The list of spectral windows and channels for which bandpasses are computed. Set to spw='' by default, which means the task will select all science spectral windows. Example: spw='11,13,15,17'
antenna	string	None	Set of data selection antenna IDs
hm_phaseup	string	snr	The pre-bandpass solution phaseup gain heuristics. The options are: 'snr': compute solution required to achieve the specified SNR 'manual': use manual solution parameters '': skip phaseup Example: hm_phaseup='manual'
phaseupsolint	any	int	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. Example: phaseupsolint='300s'
phaseupbw	string	None	Bandwidth to be used for phaseup. Used when hm_phaseup='manual'. Example: phaseupbw='' to use entire bandpass phaseupbw='500MHz' to use central 500MHz
phaseupsnr	double	20.0	The required SNR for the phaseup solution. Used only if hm_phaseup='snr'. Example: phaseupsnr=10.0
phaseupnsols	int	2	The minimum number of phaseup gain solutions. Used only if hm_phaseup='snr'. Example: phaseupnsols=4

hm_bandpass	string	snr	The bandpass solution heuristics. The options are: 'snr': compute the solution required to achieve the specified SNR 'smoothed': simple smoothing heuristics 'fixed': use the user defined parameters for all spws
solint	any	inf	Time and channel solution intervals in CASA syntax. Default is solint='inf', which is used when hm_bandpass='fixed'. If hm_bandpass is set to 'snr', then the task will attempt to compute and use an optimal SNR-based solint (and warn if this solint is not good enough). If hm_bandpass is set to 'smoothed', the task will use a smoothed solint.
maxchannels	int	240	The bandpass solution smoothing factor in channels. The solution interval is bandwidth / 240. Set to 0 for no smoothing. Used if hm_bandpass='smoothed'. Example: maxchannels=0
evenbpints	bool	True	Force the per spw frequency solint to be evenly divisible into the spw bandpass if hm_bandpass='snr'. Example: evenbpints=False
bpsnr	double	50.0	The required SNR for the bandpass solution. Used only if hm_bandpass='snr'. Example: bpsnr=30.0
minbpsnr	double	20.0	The minimum required SNR for the bandpass solution when strong atmospheric lines exist in Tsys spectra. Used only if hm_bandpass='snr'. Example: minbpsnr=10.0
bpnsols	int	8	The minimum number of bandpass solutions. Used only if hm_bandpass='snr'.
combine	string	scan	Data axes to combine for solving. Axes are '', 'scan', 'spw', 'field' or any comma-separated combination. Example: combine='scan,field'
refant	string	None	List of 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. Example: refant='DV06,DV07'
minblperant	int	4	Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions.
minsnr	double	3.0	Solutions below this SNR are rejected
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. If an iteration results in no new flags, then subsequent iterations are skipped.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Automatically accept the results of the task into the pipeline context (True) or reject them (False).

Back

---

hifa_bandpass

Task Description

Compute bandpass calibration solutions
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	stringVec	None	List of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. Example: vis=['ngc5921.ms']
caltable	stringVec	None	List of names for the output calibration tables. Defaults to the standard pipeline naming convention. Example: caltable=['ngc5921.gcal']
field	string	None	The list of field names or field ids for which bandpasses are computed. Set to field='' by default, which means the task will select all fields. Example: field='3C279', field='3C279,M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Set to intent='' by default, which means the task will select all data with the BANDPASS intent. Example: intent='*PHASE*'
spw	string	None	The list of spectral windows and channels for which bandpasses are computed. Set to spw='' by default, which means the task will select all science spectral windows. Example: spw='11,13,15,17'
antenna	string	None	Set of data selection antenna IDs
hm_phaseup	string	snr	The pre-bandpass solution phaseup gain heuristics. The options are: 'snr': compute solution required to achieve the specified SNR 'manual': use manual solution parameters '': skip phaseup Example: hm_phaseup='manual'
phaseupsolint	any	int	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. Example: phaseupsolint='300s'
phaseupbw	string	None	Bandwidth to be used for phaseup. Used when hm_phaseup='manual'. Example: phaseupbw='' to use entire bandpass phaseupbw='500MHz' to use central 500MHz
phaseupsnr	double	20.0	The required SNR for the phaseup solution. Used only if hm_phaseup='snr'. Example: phaseupsnr=10.0

phaseupnsols	int	2	The minimum number of phaseup gain solutions. Used only if hm_phaseup='snr'. Example: phaseupnsols=4
hm_bandpass	string	snr	The bandpass solution heuristics. The options are: 'snr': compute the solution required to achieve the specified SNR 'smoothed': simple smoothing heuristics 'fixed': use the user defined parameters for all spws
solint	any	inf	Time and channel solution intervals in CASA syntax. Default is solint='inf', which is used when hm_bandpass='fixed'. If hm_bandpass is set to 'snr', then the task will attempt to compute and use an optimal SNR-based solint (and warn if this solint is not good enough). If hm_bandpass is set to 'smoothed', the task will use a smoothed solint.
maxchannels	int	240	The bandpass solution smoothing factor in channels. The solution interval is bandwidth / 240. Set to 0 for no smoothing. Used if hm_bandpass='smoothed'. Example: maxchannels=0
evenbpints	bool	True	Force the per spw frequency solint to be evenly divisible into the spw bandpass if hm_bandpass='snr'. Example: evenbpints=False
bpsnr	double	50.0	The required SNR for the bandpass solution. Used only if hm_bandpass='snr'. Example: bpsnr=30.0
minbpsnr	double	20.0	The minimum required SNR for the bandpass solution when strong atmospheric lines exist in Tsys spectra. Used only if hm_bandpass='snr'. Example: minbpsnr=10.0
bpnsols	int	8	The minimum number of bandpass solutions. Used only if hm_bandpass='snr'.
combine	string	scan	Data axes to combine for solving. Axes are '', 'scan', 'spw', 'field' or any comma-separated combination. Example: combine='scan,field'
refant	string	None	List of 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. Example: refant='DV06,DV07'
solnorm	bool	True	Normalise the bandpass solution
minblperant	int	4	Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions.

minsnr	double	3.0	Solutions below this SNR are rejected
unregister_existing	bool	False	Unregister all bandpass calibrations from the pipeline context before registering the new bandpass calibrations from this task.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Automatically accept the results of the task into the pipeline context (True) or reject them (False).

Back

---

hifa_bpsolint

Task Description

Compute optimal bandpass calibration solution intervals
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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context example: vis=['M82A.ms', 'M82B.ms']
field	string	None	The list of field names of sources to be used for signal to noise estimation. Defaults to all fields with the standard intent. example: field='3C279'
intent	string	BANDPASS	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to 'BANDPASS'. example: intent='PHASE'

spw	string	None	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. example: spw='13,15'
phaseupsnr	double	20.0	The required phaseup gain time interval solution signal to noise. example: phaseupsnr=10.0
minphaseupints	int	2	The minimum number of time intervals in the phaseup gain. solution. example: minphaseupints=4
evenbpints	bool	False	
bpsnr	double	50.0	The required bandpass frequency interval solution signal to noise. example: bpsnr=30.0
minbpsnr	double	20.0	The minimum required bandpass frequency interval solution signal to noise when strong atmospheric lines exist in Tsys spectra. example: minbpsnr=10.0
minbpnchan	int	8	The minimum number of frequency intervals in the bandpass solution. example: minbpnchan=16
hm_nantennas	string	all	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. example: hm_nantennas='unflagged'
maxfracflagged	double	0.90	The maximum fraction of an antenna that can be flagged before it is excluded from the signal to noise estimate. example: maxfracflagged=0.80
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

hifa_exportdata

Task Description

Prepare interferometry data for export

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	stringVec	None	List of visibility data files for which flagging and calibration information will be exported. Defaults to the list maintained in the pipeline context. Not available in pipelinemode='automatic'. example: vis=['X227.ms', 'X228.ms']
session	stringVec	None	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. Not available in pipelinemode='automatic'. example: session=['session1', 'session2']
imaging_products_only	bool	False	Export science target imaging products only
exportmses	bool	False	Export the final MeasurementSets instead of the final flags, calibration tables, and calibration instructions.
pprfile	string	None	Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'. Not available in pipelinemode='automatic'. example: pprfile=['PPR_GRB021004.xml']
calintents	string	None	List of calibrator image types to be exported. Defaults to all standard calibrator intents, 'BANDPASS', 'PHASE', 'FLUX'. Not available in pipelinemode='automatic'. example: 'PHASE'

calimages	stringVec	None	List of calibrator images to be exported. Defaults to all calibrator images recorded in the pipeline context. Not available in pipelinemode='automatic'. example: calimages=['3C454.3.bandpass', '3C279.phase']
targetimages	stringVec	None	List of science target images to be exported. Defaults to all science target images recorded in the pipeline context. Not available in pipelinemode='automatic'. example: targetimages=['NGC3256.band3', 'NGC3256.band6']
products_dir	string	None	Name of the data products subdirectory. Defaults to './' Not available in pipelinemode='automatic'. example: '../products'
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display task command (True). Only available in pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Only available in pipelinemode='interactive'.

Back

---

hifa_flagdata

Task Description

Do meta data based flagging of a list of MeasurementSets.

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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets defined in the pipeline context.
autocorr	bool	True	Flag autocorrelation data.
shadow	bool	True	Flag shadowed antennas.
tolerance	double	0.0	Amount of antenna shadowing tolerated, in meters. A positive number allows antennas to overlap in projection. A negative number forces antennas apart in projection. Zero implies a distance of radius_1+radius_2 between antenna centers.
scan	bool	True	Flag a list of specified scans.
scannumber	string	None	A string containing a comma delimited list of scans to be flagged. example: scannumber='3,5,6'
intents	string	POINTING,FOCUS,ATMOSPHERE,SIDEBAND	A string containing a comma delimited list of intents against which the scans to be flagged are matched. example: intents='*BANDPASS*'
edgespw	bool	True	Flag the edge spectral window channels.
fracspw	double	0.03125	Fraction of channels to flag at both edges of TDM spectral windows.

fracspwfps	double	0.048387	Fraction of channels to flag at both edges of ACA TDM spectral windows that were created with the earlier (original) implementation of the frequency profile synthesis (FPS) algorithm.
online	bool	True	Apply the online flags.
partialpol	bool	True	Identify integrations in multi-polarisation data where part of the polarization products are already flagged, and flag the other polarization products in those integrations.
fileonline	string	None	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.
template	bool	True	Apply flagging templates
filetemplate	stringVec	None	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.

hm_tbuff	string	halfint	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.
tbuff	any	[0.0,0.0]	The time in seconds used to pad flagging command time intervals if hm_tbuff='manual'. The default in manual mode is no flagging.
qa0	bool	True	QA0 flags.
qa2	bool	True	QA2 flags.
pipelinemode	string	automatic	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.
flagbackup	bool	True	Back up any pre-existing flags.

dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_flagtargets

Task Description

Do science target flagging

1. Do basic flagging on a science target MeasurementSet:

```
hifa_flagtargets()
```

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets defined in the pipeline context. (can be set only in 'interactive mode')
template	bool	True	Apply flagging templates. (can be set in any pipeline mode)
filetemplate	stringVec	None	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. (can be set in any pipeline mode)
pipelinemode	string	automatic	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.
flagbackup	bool	False	Back up any pre-existing flags. (can be set only in 'interactive mode')
dryrun	bool	False	Run the commands (False) or generate the commands to be run but do not execute (True).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_fluxcalflag

Task Description

Locate and flag line regions in solar system flux calibrators

1. Locate known lines in any solar system object flux calibrators:
   hifa_fluxcalflag()

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets defined in the pipeline context.
field	string	None	The list of field names or field ids for which the models are to be set. Defaults to all fields with intent 'AMPLITUDE'. example: field='3C279', field='3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to all data with amplitude intent. example: intent='AMPLITUDE'
spw	string	None	The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows. example: spw='11,13,15,17'
pipelinemode	string	automatic	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.
threshold	double	0.75	If the fraction of an spw occupied by line regions is greater then threshold flag the entire spectral window.
appendlines	bool	False	Append user defined line regions to the line dictionary.
linesfile	string	None	
applyflags	bool	True	
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_gaincalsnr

Task Description

Compute gaincal signal to noise ratios per spw
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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. example: vis=['M82A.ms', 'M82B.ms']
field	string	None	The list of field names of sources to be used for signal to noise estimation. Defaults to all fields with the standard intent. example: field='3C279'
intent	string	PHASE	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to 'PHASE'. example: intent='BANDPASS'
spw	string	None	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. example: spw='13,15'
phasesnr	double	25.0	The required gaincal solution signal to noise. example: phasesnr=20.0
bwedgefrac	double	0.03125	The fraction of the bandwidth edges that is flagged. example: bwedgefrac=0.0
hm_nantennas	string	all	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. example: hm_nantennas='unflagged'
maxfracflagged	double	0.90	The maximum fraction of an antenna that can be flagged before it is excluded from the signal to noise estimate. example: maxfracflagged=0.80
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

hifa_gfluxscaleflag

Task Description

Derive the flux density scale with flagging
1. run with recommended settings to create flux scale calibration with flagging
using recommended thresholds:
hifa_gfluxscaleflag()

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. example: vis=['M51.ms']
intent	string	None	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 AMPLITUDE, PHASE, and CHECK intents, except for one case: if one of the AMPLITUDE intent fields was also used for BANDPASS, then this task will select only data with PHASE and CHECK intents. example: intent='*PHASE*'
phaseupsolint	any	int	The phase correction solution interval in CASA syntax. example: phaseupsolint='300s'
solint	any	inf	Time and channel solution intervals in CASA syntax. example: solint='inf,10ch', solint='inf'
minsnr	double	2.0	Solutions below this SNR are rejected.
refant	string	None	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. example: refant='DV01', refant='DV06,DV07'
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. If an iteration results in no new flags, then subsequent iterations are skipped.
pipelinemode	string	automatic	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'.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_gfluxscale

Task Description

Derive flux density scales from standard calibrators
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	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context example: ['M32A.ms', 'M32B.ms']
reference	variant	None	A string containing a comma delimited list of field names defining the reference calibrators. Defaults to field names with intent '*AMP*'. example: reference='M82,3C273'
transfer	variant	None	A string containing a comma delimited list of field names defining the transfer calibrators. Defaults to field names with intent '*PHASE*'. example: transfer='J1328+041,J1206+30'
refintent	string	None	A string containing a comma delimited list of intents used to select the reference calibrators. Defaults to 'AMPLITUDE'. example: refintent='', refintent='AMPLITUDE'
transintent	string	None	A string containing a comma delimited list of intents defining the transfer calibrators. Defaults to 'PHASE,BANDPASS,CHECK,POLARIZATION,POLANGLE,POLLEAKAGE'. example: transintent='', transintent='PHASE,BANDPASS'
refspwmap	intVec	None	Vector of spectral window ids enabling scaling across spectral windows. Defaults to no scaling. example: refspwmap=[1,1,3,3] - (4 spws, reference fields in 1 and 3, transfer fields in 0,1,2,3
reffile	string	None	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. example: reffile='', reffile='working/flux.csv'
phaseupsolint	any	int	Time solution intervals in CASA syntax for the phase solution. example: phaseupsolint='inf', phaseupsolint='int', phaseupsolint='100sec'
solint	any	inf	Time solution intervals in CASA syntax for the amplitude solution. example: solint='inf', solint='int', solint='100sec'
minsnr	double	2.0	Minimum signal to noise ratio for gain calibration solutions. example: minsnr=1.5, minsnr=0.0
refant	string	None	A string specifying the reference antenna(s). By default this is read from the context. example: refant='DV05'
hm_resolvedcals	string	automatic	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	string	None	A comma delimited string specifying the antenna names or ids to be used for the fluxscale determination. Used in hm_resolvedcals='manual' mode. example: antenna='DV16,DV07,DA12,DA08'
peak_fraction	double	0.2	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.

pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_imageprecheck

Task Description

Calculates the best robust value and Briggs weighting parameter to achieve sensitivity and angular resolution goals.

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
parallel	string	automatic	Use MPI cluster where possible
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_importdata

Task Description

Imports data into the interferometry pipeline

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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
session	stringVec	None	List of visibility data sessions
pipelinemode	string	automatic	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.

asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR CalPointing	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. Can only be set in pipelinemode='interactive'. When converting ASDM to MS, if overwrite=False and the MS already exists in output directory, then this existing MS dataset will be used instead.
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	False	Use the online flux catalog
ocorr_mode	string	ca	ALMA default set to ca
createmms	string	false	Create an MMS
minparang	double	0.0	Minimum required parallactic angle range for polarisation calibrator, in degrees. The default of 0.0 is used for non-polarisation processing.
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results into the pipeline context

Back

---

hifa_lock_refant

Task Description

Lock reference antenna list

Parameter List

No parameter

Back

---

hifa_polcalflag

Task Description

Flag polarization calibrators

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_renorm

Task Description

Base renorm task

Parameter List

name	type	default	description
vis	stringVec	None	List of input visibility data
apply	bool	False	Apply renormalization correction
threshold	float	1.02	Apply correction if max correction is above threshold and apply=True
correctATM	bool	False	Use the ATM model transmission profiles to try correct for any ATM residual features that get into the scaling spectra
spw	string	None	The list of spectral windows. Set to spw='' by default, which means the task will select all spectral windows. Example: spw='11,13,15,17'
excludechan	string	None	Channels to exclude, e.g. excludechan="{'22':'100~150'}"
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_restoredata

Task Description

Restore flagged and calibration interferometry data from a pipeline run

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	stringVec	None	List of raw visibility data files to be restored. Assumed to be in the directory specified by rawdata_dir. (can be set only in 'interactive mode') example: vis=['uid___A002_X30a93d_X43e']
session	stringVec	None	List of sessions one per visibility file. (can be set only in 'interactive mode') example: session=['session_3']

products_dir	string	../products	Name of the data products directory to copy calibration products from. The parameter is effective only when copytoraw = True. When copytoraw = False, calibration products in rawdata_dir will be used. (can be set only in 'interactive mode') example: products_dir='myproductspath'
copytoraw	bool	True	Copy calibration and flagging tables from products_dir to rawdata_dir directory. (can be set only in 'interactive mode') example: copytoraw=False
rawdata_dir	string	../rawdata	Name of the rawdata subdirectory. (can be set only in 'interactive mode') example: rawdata_dir='myrawdatapath'
lazy	bool	False	Use the lazy filler option. (can be set only in 'interactive mode') example: lazy=True
bdfflags	bool	True	Set the BDF flags. (can be set only in 'interactive mode') example: bdfflags=False
ocorr_mode	string	ca	Set ocorr_mode. (can be set only in 'interactive mode') example: ocorr_mode='ca'
asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR CalPointing	Set list of tables to import as is. (can be set only in 'interactive mode') example: asis='Source Receiver'
pipelinemode	string	automatic	The pipeline operating mode. In 'automatic' mode the pipeline determines the values of all context defined pipeline inputs automatically. (can be set in any pipeline mode)
dryrun	bool	False	Run the commands (False) or generate the commands to be run but do not execute (True).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

hifa_session_bandpass

Task Description

Compute bandpass calibration solutions
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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. example: vis=['M51.ms']
caltable	stringVec	None	The list of output calibration tables. Defaults to the standard pipeline naming convention. example: caltable=['M51.bcal']
field	string	None	The list of field names or field ids for which bandpasses are computed. Defaults to all fields. example: field='3C279', field='3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to all data with bandpass intent. example: intent='*PHASE*'
spw	string	None	The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows. example: spw='11,13,15,17'
antenna	string	None	
hm_phaseup	string	snr	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). example: hm_phaseup='manual'
phaseupsolint	any	int	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. example: phaseupsolint='300s'
phaseupbw	string	None	Bandwidth to be used for phaseup. Defaults to 500MHz. Used when hm_phaseup='manual'. example: phaseupbw='' to use entire bandpass, phaseupbw='500MHz' to use central 500MHz

phaseupsnr	double	20.0	The required SNR for the phaseup solution. Used only if hm_phaseup='snr'. example: phaseupsnr=10.0
phaseupnsols	int	2	The minimum number of phaseup gain solutions. Used only if hm_phaseup='snr'. example: phaseupnsols=4
hm_bandpass	string	snr	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	any	inf	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	int	240	The bandpass solution smoothing factor in channels. The solution interval is bandwidth / 240. Set to 0 for no smoothing. Used if hm_bandpass='smoothed". example: 0
evenbpints	bool	True	Force the per spw frequency solint to be evenly divisible into the spw bandpass if hm_bandpass='snr'. example: evenbpints=False
bpsnr	double	50.0	The required SNR for the bandpass solution. Used only if hm_bandpass='snr' example: bpsnr=30.0
minbpsnr	double	20.0	The minimum required SNR for the bandpass solution when strong atmospheric lines exist in Tsys spectra. Used only if hm_bandpass='snr'. example: minbpsnr=10.0
bpnsols	int	8	The minimum number of bandpass solutions. Used only if hm_bandpass='snr'.
hm_bandtype	string	channel	The type of bandpass. The options are 'channel' and 'polynomial' for CASA bandpass types = 'B' and 'BPOLY' respectively.
combine	string	scan	Data axes to combine for solving. Axes are '', 'scan', 'spw', 'field' or any comma-separated combination. example: combine='scan,field'
refant	string	None	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. example: refant='DV01', refant='DV06,DV07'
solnorm	bool	True	Normalise the bandpass solutions.
minblperant	int	4	Minimum number of baselines required per antenna for each solve Antennas with fewer baselines are excluded from solutions. Used for hm_bandtype='channel' only.

minsnr	double	3.0	Solutions below this SNR are rejected. Used for hm_bandtype= ’channel’ only.
degamp	variant	None	
degphase	variant	None	
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).
parallel	string	automatic	Execute using CASA HPC functionality, if available.

Back

---

hifa_session_refant

Task Description

Select best reference antenna for session(s)

1. Compute a single common reference antenna per session:
   hifa_session_refant()

Parameter List

name	type	default	description
vis	stringVec	None	List of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. Example: vis=['ngc5921.ms']
phase_threshold	double	0.005	Threshold (in degrees) used to identify absolute phase solution outliers in caltables. Example: phase_threshold=0.005
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Automatically accept the results of the task into the pipeline context (True) or reject them (False).

Back

---

hifa_spwphaseup

Task Description

Compute phase calibration spw map and per spw phase offsets
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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. example: vis=['M82A.ms', 'M82B.ms']
caltable	stringVec	None	The list of output calibration tables. Defaults to the standard pipeline naming convention. example: caltable=['M82.gcal', 'M82B.gcal']

field	string	None	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. example: field='3C279', field='3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to the BANDPASS observations. example: intent='PHASE'
spw	string	None	The list of spectral windows and channels for which gain solutions are computed. Defaults to all the science spectral windows. example: spw='13,15'
hm_spwmapmode	string	auto	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. example: hm_spwmapmode='combine'
maxnarrowbw	string	300MHz	The maximum bandwidth defining narrow spectral windows. Values must be in CASA compatible frequency units. example: maxnarrowbw=''
minfracmaxbw	double	0.8	The minimum fraction of the maximum bandwidth in the set of spws to use for matching. example: minfracmaxbw=0.75
samebb	bool	True	Match within the same baseband if possible. example: samebb=False
phasesnr	double	32.0	The required gaincal solution signal to noise. example: phaseupsnr=20.0
bwedgefrac	double	0.03125	The fraction of the bandwidth edges that is flagged. example: bwedgefrac=0.0
hm_nantennas	string	all	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. example: hm_nantennas='unflagged'
maxfracflagged	double	0.90	The maximum fraction of an antenna that can be flagged before it is excluded from the signal to noise estimate. example: maxfracflagged=0.80

combine	string	None	Data axes to combine for solving. Options are ’’, ’scan’, ’spw’, ’field’ or any comma-separated combination. example: combine=’’
refant	string	None	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. example: refant=’DV01’, refant=’DV05,DV07’
minblperant	int	4	Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions. example: minblperant=2
minsnr	double	3.0	Solutions below this SNR are rejected.
unregister_existing	bool	False	Unregister previous spwphaseup calibrations from the pipeline context before registering the new calibrations from this task.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_targetflag

Task Description

Flag target source outliers

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display task command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_timegaincal

Task Description

Determine temporal gains from calibrator observations
1. Compute standard per scan gain solutions that will be used to calibrate
the target:
hifa_timegaincal()

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the pipeline context. example: vis=['M82A.ms', 'M82B.ms']
calamptable	stringVec	None	The list of output diagnostic calibration amplitude tables for the calibration targets. Defaults to the standard pipeline naming convention. example: calamptable=['M82.gacal', 'M82B.gacal']
offsetstable	stringVec	None	The list of output diagnostic phase offset tables for the calibration targets. Defaults to the standard pipeline naming convention. example: offsetstable=['M82.offsets.gacal', 'M82B.offsets.gacal']

calphasetable	stringVec	None	The list of output calibration phase tables for the calibration targets. Defaults to the standard pipeline naming convention. example: calphasetable=['M82.gcal', 'M82B.gcal']
targetphasetable	stringVec	None	The list of output phase calibration tables for the science targets. Defaults to the standard pipeline naming convention. example: targetphasetable=['M82.gcal', 'M82B.gcal']
amptable	stringVec	None	The list of output calibration amplitude tables for the calibration and science targets. Defaults to the standard pipeline naming convention. example: amptable=['M82.gcal', 'M82B.gcal']
field	string	None	The list of field names or field ids for which gain solutions are to be computed. Defaults to all fields with the standard intent. example: field='3C279', field='3C279, M82'
intent	string	None	A string containing a comma delimited list of intents against which the selected fields are matched. Defaults to the equivalent of 'AMPLITUDE,PHASE,BANDPASS'. example: intent='', intent='PHASE'
spw	string	None	The list of spectral windows and channels for which gain solutions are computed. Defaults to all science spectral windows. example: spw='3C279', spw='3C279, M82'
antenna	string	None	None
calsolint	any	int	Time solution interval in CASA syntax for calibrator source solutions. example: calsolint='inf', calsolint='int', calsolint='100sec'
targetsolint	any	inf	Time solution interval in CASA syntax for target source solutions. example: targetsolint='inf', targetsolint='int', targetsolint='100sec'
combine	string	None	Data axes to combine for solving. Options are '', 'scan', 'spw', 'field' or any comma-separated combination. default: '' example: combine=''
refant	string	None	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. example: refant='DV01', refant='DV05,DV07'
refantmode	string	None	Controls how the refant is applied. Currently available choices are 'flex', 'strict', and the default value of ''. Setting to '' allows the pipeline to select the appropriate mode based on the state of the reference antenna list. Examples: refantmode='strict', refantmode=''

solnorm	bool	False	Normalise the gain solutions.
minblperant	int	4	Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions. example: minblperant=2
calminsnr	double	2.0	Solutions below this SNR are rejected for calibrator solutions.
targetminsnr	double	3.0	Solutions below this SNR are rejected for science target solutions.
smodel	doubleVec	None	Point source Stokes parameters for source model (experimental) Defaults to using standard MODEL_DATA column data. example: smodel=[1,0,0,0] - (I=1, unpolarized)
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_tsysflag

Task Description

Flag deviant system temperatures for ALMA interferometry measurements. This is done by running a sequence of flagging subtasks, each looking for a different type of possible error.

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	stringVec	None	List of input MeasurementSets (Not used).

caltable	stringVec	None	List of input Tsys calibration tables. default: [] - Use the table currently stored in the pipeline context. example: caltable=['X132.ms.tsys.s2.tbl']
flag_nmedian	bool	True	True to flag Tsys spectra with high median value.
fnm_limit	double	2.0	Flag spectra with median value higher than fnm_limit * median of this measure 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	13.0	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 > 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 that provides the reference shape for 'flag_fieldshape'.
ff_max_limit	double	13	Flag Tsys spectra with 'fieldshape' metric values > 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 > fb_sharps_limit.

flag_toomany	bool	True	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.
tmf1_limit	double	0.666	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.
tmef1_limit	double	0.666	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.
metric_order	string	nmedian,derivative,edgechans,fieldshape,birdies,toomany	Order in which to evaluate the flagging metrics that are enables. Disabled metrics are skipped.
normalize_tsys	bool	False	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.
filetemplate	stringVec	None	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.

pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hifa_unlock_refant

Task Description

Unlock reference antenna list

Parameter List

No parameter

Back

---

hifa_wvrgcalflag

Task Description

Generate a gain table based on Water Vapor Radiometer data, interpolating over antennas with bad radiometers.

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

Parameter List

name	type	default	description

vis	stringVec	None	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	stringVec	None	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	stringVec	None	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	string	automatic	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.
toffset	double	0	Time offset (sec) between interferometric and WVR data.
segsource	bool	True	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.
sourceflag	stringVec	None	Flag the WVR data for these source(s) as bad and do not produce corrections for it. Requires segsource=True. example: sourceflag=['3C273']
hm_tie	string	automatic	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.

tie	stringVec	None	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'. example: tie=['3C273,NGC253', 'IC433,3C279']
nsol	int	1	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').
disperse	bool	False	Apply correction for dispersion.
wvrflag	stringVec	None	Flag the WVR data for these antenna(s) as bad and replace its data with interpolated values. example: wvrflag=['DV03','DA05','PM02']
hm_smooth	string	automatic	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	string	None	Smooth WVR data on this timescale before calculating the correction. Ignored unless hm_smooth='manual'.
scale	double	1	Scale the entire phase correction by this factor.
maxdistm	double	-1	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	int	2	Minimum number of nearby antennas (up to 3) used for interpolation from a flagged antenna. example: minnumants=3

mingoodfrac	double	0.8	Minimum fraction of good data per antenna. example: mingoodfrac=0.7
refant	string	None	Ranked comma delimited list of reference antennas. example: refant='DV02,DV06'
flag_intent	string	None	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.
qa_intent	string	BANDPASS,PHASE	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.

qa_bandpass_intent	string	None	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.
accept_threshold	double	1.0	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.
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	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. example: ants_with_wvr_thresh=0.5
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 Water Vapor Radiometer data, and calculate
a QA score based on its effect on the interferometric data.
Example
1. Compute the WVR calibration for all the MeasurementSets:
hifa_wvrgcal(hm_tie='automatic')

Parameter List

name	type	default	description
vis	stringVec	None	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	stringVec	None	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	stringVec	None	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	string	automatic	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.
toffset	double	0	Time offset (sec) between interferometric and WVR data.
segsource	bool	True	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.
sourceflag	stringVec	None	Flag the WVR data for these source(s) as bad and do not produce corrections for it. Requires segsource=True. example: ['3C273']
hm_tie	string	automatic	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.
tie	stringVec	None	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'. example: ['3C273,NGC253', 'IC433,3C279']

nsol	int	1	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').
disperse	bool	False	Apply correction for dispersion.
wvrflag	stringVec	None	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. example: ['DV03','DA05','PM02']
hm_smooth	string	automatic	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	string	None	Smooth WVR data on this timescale before calculating the correction. Ignored unless hm_smooth='manual'.
scale	double	1.	Scale the entire phase correction by this factor.
maxdistm	double	-1	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	int	2	Minimum number of nearby antennas (up to 3) used for interpolation from a flagged antenna. example: minnumants=3
mingoodfrac	double	0.8	Minimum fraction of good data per antenna.
refant	string	None	Ranked comma delimited list of reference antennas. example: refant='DV01,DV02'

qa_intent	string	None	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=''. example: qa_intent='PHASE'
qa_bandpass_intent	string	None	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.
qa_spw	string	None	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.
accept_threshold	double	1.0	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.
pipelinemode	string	automatic	None
dryrun	bool	False	None
acceptresults	bool	True	None

Back

Summary of ALMA-specific SRDP interferometric tasks and parameters

hifas_imageprecheck

Task Description

Calculates the best robust value and Briggs weighting parameter to achieve sensitivity and angular resolution goals.

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets specified in the h_init or hif_importdata task. '': use all MeasurementSets in the context Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
desired_angular_resolution	string	None	User specified angular resolution goal string. '': automatic from performance parameters (default) Example: '1.0arcsec'
calcsb	bool	False	Force (re-)calculation of sensitivities and beams
parallel	string	automatic	Use MPI cluster where possible
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or just display the command (True)
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

Summary of VLA-specific interferometric tasks and parameters

hifv_applycals

Task Description

Applycals
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
field	string	None	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. Only can be set in pipelinemode='interactive'. example: '3C279', '3C279, M82'
intent	string	None	A string containing the list of intents against which the selected fields will be matched. Defaults to all supported intents in the pipeline context. Only can be set in pipelinemode='interactive'. example: '*TARGET*'
spw	string	None	The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline. Only can be set in pipelinemode='interactive'. example: '17', '11, 15'

antenna	string	None	The list of antennas to which the calibration will be applied. Defaults to all antennas. Not currently supported. Only can be set in pipelinemode='interactive'.
applymode	string	None	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)
flagbackup	bool	True	Backup the flags before the apply. Only can be set in pipelinemode='interactive'.
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. 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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_checkflag

Task Description

Run flagdata in rflag mode

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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
checkflagmode	string	None	-- 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 use 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. -- Mode 'vlass-imaging' is similar to 'target-vlass', except that it executes on the split off target data, intent='*TARGET', datacolumn='data' and uses a timedevscale of 4.0. -- Standard VLA modes with improved RFI flaggig heuristics: 'bpd-vla', 'allcals-vla', 'target-vla'
growflags	bool	True	Grow flags in time at the end of the following checkflagmodes: default=True, for 'bpd-vla', 'allcals-vla', 'bpd', and 'allcals' default=False, for '' and 'semi'

pipelinemode	string	automatic	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.
overwrite_modelcol	bool	False	Always write the model column, even if it already exists
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_circfeedpolcal

Task Description

Base circfeedpolcal task
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	stringVec	None	List of input visibility data
Dterm_solint	string	2MHz	D-terms spectral averaging. Example: refantignore='ea02,ea03'
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	doubleVec	0.25	Range to use for flag clipping
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_exportdata

Task Description

Prepare interferometry data for export
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	stringVec	None	List of visibility data files for which flagging and calibration information will be exported. Defaults to the list maintained in the pipeline context. Not available in pipelinemode='automatic'. example: vis=['X227.ms', 'X228.ms']
session	stringVec	None	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. Not available in pipelinemode='automatic'. example: session=['session1', 'session2']
imaging_products_only	bool	False	Export science target imaging products only
exportmses	bool	False	Export the final MeasurementSets instead of the final flags, calibration tables, and calibration instructions.
exportcalprods	bool	False	Export flags and caltables in addition to MeasurementSets. this parameter is only valid when exportmses = True.
pprfile	string	None	Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'. Not available in pipelinemode='automatic'. example: pprfile=['PPR_GRB021004.xml']
calintents	string	None	List of calibrator image types to be exported. Defaults to all standard calibrator intents, 'BANDPASS', 'PHASE', 'FLUX'. Not available in pipelinemode='automatic'. example: 'PHASE'
calimages	stringVec	None	List of calibrator images to be exported. Defaults to all calibrator images recorded in the pipeline context. Not available in pipelinemode='automatic'. example: calimages=['3C454.3.bandpass', '3C279.phase']
targetimages	stringVec	None	List of science target images to be exported. Defaults to all science target images recorded in the pipeline context. Not available in pipelinemode='automatic'. example: targetimages=['NGC3256.band3', 'NGC3256.band6']
products_dir	string	None	Name of the data products subdirectory. Defaults to './' Not available in pipelinemode='automatic'. example: '../products'

gainmap	bool	False	The value of gainmap parameter in hifv_restoredata task put in casa_piperestorescript.py
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display task command (True). Only available in pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Only available in pipelinemode='interactive'.

Back

---

hifv_exportvlassdata

Task Description

Base exportvlassdata task

Examples

1. Basic exportvlassdata task

   hifv_exportvlassdata()

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_finalcals

Task Description

Finalcals
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.
weakbp	bool	False	Activate weak bandpass heuristics
refantignore	string	None	String list of antennas to ignore

Back

---

hifv_fixpointing

Task Description

Base fixpointing task
The hifv_fixpointing task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visisbility 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 fixpointing task
hifv_fixpointing()

Parameter List

name	type	default	description
vis	stringVec	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_flagbaddef

Task Description

Flagging of deformatters - amp and phase
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
doflagundernspwlimit	bool	False	If the number of bad spws is greater than zero, and the keyword is True, then spws are flagged individually.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_flagcal

Task Description

Flagcal task
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	stringVec	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. 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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_flagdata

Task Description

Do basic deterministic flagging of a list of MeasurementSets
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.

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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
autocorr	bool	True	Flag autocorrelation data
shadow	bool	True	Flag shadowed antennas
scan	bool	True	Flag specified scans
scannumber	string	None	A string containing a comma delimited list of scans to be flagged. example: '3,5,6'
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*	A string containing a comma delimited list of intents against which the scans to be flagged are matched. example: '*BANDPASS*'
edgespw	bool	True	Fraction of the baseline correlator TDM edge channels to be flagged.
fracspw	double	0.05	Fraction of baseline correlator edge channels to be flagged
online	bool	True	Apply the online flags
fileonline	string	None	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.
template	bool	True	Apply a flagging template

filetemplate	stringVec	None	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.
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. 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.
flagbackup	bool	False	Backup pre-existing flags before applying new ones. Only can be set in pipelinemode='interactive'.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

hifv_fluxboot

Task Description

Fluxboot
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
caltable	string	None	String name of the flagged caltable
fitorder	int	-1	Polynomial order of the spectral fitting for valid flux densities with multiple spws. The default value of -1 means that the heuristics determine the fit order based on fractional bandwidth and receiver bands present in the observation. An override value of 0,1,2,3 or 4 may be specified by the user. Spectral index (1) and, if applicable, curvature (2) are reported in the weblog. If no determination can be made by the heuristics, a fitorder of 1 will be used.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.
refantignore	string	None	String list of antennas to ignore Example: refantignore='ea02,ea03'

Back

hifv_gaincurves

Task Description

Runs gencal in gc mode

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	stringVec	None	List of input visibility data
caltable	string	None	String name of caltable
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

hifv_hanning

Task Description

Hanning smoothing on a dataset
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

hifv_importdata

Task Description

Imports data into the VLA pipeline
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
session	stringVec	None	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. example: session=['Session_1', 'Sessions_2']
pipelinemode	string	automatic	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.
asis	string	Receiver CalAtmosphere	ASDM to convert as is Only can be set in pipelinemode='interactive' examples: 'Receiver CalAtmosphere' 'Receiver', ''
overwrite	bool	False	Only can be set in pipelinemode='interactive'
nocopy	bool	False	When importing an MS, disable copying of the MS to the working directory. Only can be set in pipelinemode='interactive'
createmms	string	automatic	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.
ocorr_mode	string	co	Read in cross- and auto-correlation data(ca), cross-correlation data only (co), or autocorrelation data only (ao).
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

hifv_opcal

Task Description

Runs gencal in opac mode
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	stringVec	None	List of input visibility data
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_pbcor

Task Description

Base pbcor task
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	stringVec	None	List of input visibility data
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_plotsummary

Task Description

End of VLA pipeline plotsummary
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	stringVec	None	List of input visibility data
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_priorcals

Task Description

Runs gaincurves, opacities, requantizer gains, antenna position corrections, tec_maps, switched power.
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
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. 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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_restoredata

Task Description

Restore flagged and calibration interferometry data from a pipeline run

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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
session	stringVec	None	List of sessions one per visibility file. Only can be set in pipelinemode='interactive'. Example: session=['session_3']

products_dir	string	../products	Name of the data products directory to copy calibration products from. The parameter is effective only when copytoraw = True When copytoraw = False, calibration products in rawdata_dir will be used. Only can be set in pipelinemode='interactive'. example: products_dir='myproductspath'
copytoraw	bool	True	Copy calibration and flagging tables from products_dir to rawdata_dir directory. Only can be set in pipelinemode='interactive'. Example: copytoraw=False.
rawdata_dir	string	../rawdata	The rawdata directory. Only can be set in pipelinemode='interactive'. Example: rawdata_dir='myrawdatapath'
lazy	bool	False	Use the lazy filler option. Only can be set in pipelinemode='interactive'.
bdfflags	bool	False	Set the BDF flags. Only can be set in pipelinemode='interactive'.
ocorr_mode	string	co	Correlation import mode
gainmap	bool	False	If True, map gainfields to a particular list of scans when applying calibration tables.
asis	string	None	List of tables to import asis. Only can be set in pipelinemode='interactive'.
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_rqcal

Task Description

Runs gencal in rq mode
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	stringVec	None	List of input visibility data
caltable	string	None	String name of caltable
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_selfcal

Task Description

Selfcal task executing gaincal and applycal

Examples

1. Basic selfcal task
   hifv_selfcal()
2. VLASS-SE selfcal usage
   hifv_selfcal(selfcalmode='VLASS-SE', combine='field,spw')

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.
refantignore	string	None	String list of antennas to ignore
combine	string	spw,field	Data axes which to combine for solve Options: '','obs','scan','spw',field', or any comma-separated combination in a single string Example: combine='scan,spw' - Extend solutions over scan boundaries (up to the solint), and combine spws for solving. In selfcalmode='VLASS-SE' use the default value.
selfcalmode	string	VLASS	Heuristics mode selection. Known modes are 'VLASS' and 'VLASS-SE'. Default value is 'VLASS'.
refantmode	string	strict	Reference antenna mode
overwrite_modelcol	bool	False	Always write the model column, even if it already exists

Back

---

hifv_semiFinalBPdcals

Task Description

Runs a second quick calibration to set up for heuristic flagging

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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.
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
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.
limit_short_solint	any	None	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'.
refantignore	string	None	String list of antennas to ignore Example: refantignore='ea02,ea03'

Back

---

hifv_statwt

Task Description

Statwt
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()
2. Statistical weighting of the visibilities in the Very Large Array Sky Survey Single Epoch use case:
hifv_statwt(mode='vlass-se', datacolumn='residual_data')

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
datacolumn	string	corrected	Data column used to compute weights. Supported values are "data", "corrected", "residual", and "residual_data" (case insensitive, minimum match supported).
pipelinemode	string	automatic	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.
overwrite_modelcol	bool	False	Always write the model column, even if it already exists
statwtmode	string	VLA	Sets the weighting parameters for general VLA ('VLA') or VLASS Single Epoch ('VLASS-SE') use case. Note that the 'VLASS-SE' mode is meant to be used with datacolumn='residual_data'. Default is 'VLA'.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_swpowcal

Task Description

Runs gencal in swpow mode
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	stringVec	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. 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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_syspower

Task Description

Base syspower task
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	stringVec	None	List of input visibility data
clip_sp_template	doubleVec	1.2	Clipping range
antexclude	string	None	csv string list of antennas to exclude
usemedian	bool	False	If antexclude is specified with usemedian=False, the template values are replaced with 1.0. If usemedian = True, the template values are replaced with the median of the good antennas
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_targetflag

Task Description

Targetflag
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.
intents	string	*CALIBRATE*,*TARGET*	List of intents of scans to be flagged

Back

---

hifv_tecmaps

Task Description

Base tecmaps task
Examples
1. Basic tecmaps task
hifv_tecmaps()

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

---

hifv_testBPdcals

Task Description

Runs initial delay calibration to set up heuristic flagging
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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.
weakbp	bool	False	Activate weak bandpass heuristics
refantignore	string	None	String list of antennas to ignore Example: refantignore='ea02,ea03'
doflagundernspwlimit	bool	False	If the number of bad spws is greater than zero, and the keyword is True, then spws are flagged individually.

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
Output:
results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
the results object for the pipeline task is returned.
standard -- Flux density standard
default: ''
Examples
1. Initial run of setjy:
hifv_vlasetjy()

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
field	string	None	List of field names or ids. Only can be set in pipelinemode='interactive'.
intent	string	None	Observing intent of flux calibrators. Only can be set in pipelinemode='interactive'.
spw	string	None	List of spectral window ids. Only can be set in pipelinemode='interactive'.
model	string	None	File location for field model. Only can be set in pipelinemode='interactive'.
reffile	string	None	Path to file with fluxes for non-solar system calibrators. Only can be set in pipelinemode='interactive'
fluxdensity	any	-1	Specified flux density [I,Q,U,V]; -1 will lookup values
spix	double	0.0	Spectral index of fluxdensity. Can be set when fluxdensity is not -1
reffreq	string	1GHz	Reference frequency for spix. Can be set when fluxdensity is not -1
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. 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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

hifv_vlassmasking

Task Description

Base vlassmasking task
The hifv_vlassmasking task
Keyword arguments:
---- pipeline parameter arguments which can be set in any pipeline mode
vis -- List of visisbility 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 vlassmasking task
hifv_vlassmasking()

Parameter List

name	type	default	description
vis	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
vlass_ql_database	string	None	vlass_ql_database - usage in Socorro: /home/vlass/packages/VLASS1Q.fits
maskingmode	string	vlass-se-tier-1	maskingmode options are vlass-se-tier-1 or vlass-se-tier-2
catalog_search_size	any	1.5	catalog_search_size in units of degrees
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False). This is a pipeline task execution mode.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). This is a pipeline task execution mode.

Back

Created at Wed Sep 8 10:27:27 2021 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
1. Apply the calibration to the target data
hsd_applycal (intent='TARGET')
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.

Parameter List

name	type	default	description
vis	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets in the pipeline context. Parameter not available in pipelinemode='automatic'. example: ['X227.ms']
field	string	None	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. Parameter not available in pipelinemode='automatic'. example: '3C279', '3C279, M82'
intent	string	None	A string containing the list of intents against which the selected fields will be matched. Defaults to all supported intents in the pipeline context. Parameter not available in pipelinemode='automatic'. example: '*TARGET*'
spw	string	None	The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context. Parameter not available in pipelinemode='automatic'. example: '17', '11, 15'
antenna	string	None	The list of antennas to which the calibration will be applied. Defaults to all antennas. Not currently supported.
applymode	string	None	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)
calwt	boolVec	True	Calibrate the weights as well as the data. Parameter not available in pipelinemode='automatic'.
flagbackup	bool	True	Backup the flags before the apply

pipelinemode	string	automatic	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.
dryrun	bool	False	Run task (False) or display the command(True). Available only when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Available only when pipelinemode='interactive'.

Back

---

hsd_atmcor

Task Description

Apply offline ATM correction to the data.

Parameter List

name	type	default	description
atmtype	any	1	Type of atmospheric transmission model represented as an integer. Available options are, 1: tropical (default) 2: mid latitude summer 3: mid latitude winter 4: subarctic summer 5: subarctic winter
dtem_dh	any	-5.6K/km	temperature gradient [K/km], e.g. -5.6. ("" = Tool default) The value is directly passed to initialization method for ATM model. Float and string types are acceptable. Float value is interpreted as the value in K/km. String value should be the numeric value with unit such as '-5.6K/km'. Default: '' (tool default, -5.6K/km, is used)
h0	any	2km	scale height for water [km], e.g. 2.0. ("" = Tool default) The value is directly passed to initialization method for ATM model. Float and string types are acceptable. Float value is interpreted as the value in kilometer. String value should be the numeric value with unit compatible with length, such as '2km' or '2000m'.

pipelinemode	string	automatic	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.
infiles	stringVec	None	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_atmcor.
antenna	string	None	Data selection by antenna names or ids. example: 'PM03,PM04' '' (all antennas)
field	string	None	Data selection by field names or ids. example: '*Sgr*,M100' '' (all fields)
spw	string	None	Data selection by spw ids. example: '3,4' (spw 3 and 4) '' (all spws)
pol	string	None	Data selection by polarizations. example: 'XX,YY' (correlation XX and YY) '' (all polarizations)
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).
parallel	string	automatic	Execute using CASA HPC functionality, if available.

Back

---

hsd_baseline

Task Description

Detect and validate spectral lines, subtract baseline by masking detected lines

Parameter List

name	type	default	description
fitfunc	string	cspline	fitting function for baseline subtraction. You can only choose cubic spline ('spline' or 'cspline')
fitorder	int	-1	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.

switchpoly	bool	True	If True, switch to 1st or 2nd order polynomial fit when large mask exists at edge regardless of whatever fitfunc or fitorder are specified. Condition for switching is as follows: if nmask > nchan/2 => 1st order polynomial else if nmask > nchan/4 => 2nd order polynomial else => use fitfunc and fitorder where nmask is a number of channels for mask at edge while nchan is a number of channels of entire spectral window.

linewindow	any	None	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) example: [100,200] (channel), [115e9, 115.1e9] (frequency in Hz) ['115GHz', '115.1GHz'], see above for more examples

linewindowmode	string	replace	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.
edge	intVec	None	Number of edge channels to be dropped from baseline subtraction. The value must be a list with length of 2, whose values specify left and right edge channels, respectively. example: [10,10]
broadline	bool	True	Try to detect broad component of spectral line if True.
clusteringalgorithm	string	hierarchy	Selection of the algorithm used in the clustering analysis to check the validity of detected line features. 'kmean' algorithm and hierarchical clustering algorithm 'hierarchy', and their combination ('both') are so far implemented.
deviationmask	bool	True	Apply deviation mask in addition to masks determined by the automatic line detection.
pipelinemode	string	automatic	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.
infiles	stringVec	None	List of data files. These must be a name of MeasurementSets that are registered to context via hsd_importdata task. example: vis=['X227.ms', 'X228.ms']
field	string	None	Data selection by field. example: '1' (select by FIELD_ID) 'M100*' (select by field name) '' (all fields)
antenna	string	None	Data selection by antenna. example: '1' (select by ANTENNA_ID) 'PM03' (select by antenna name) '' (all antennas)
spw	any	None	Data selection by spw. example: '3,4' (generate caltable for spw 3 and 4) ['0','2'] (spw 0 for first data, 2 for second) '' (all spws)

pol	any	None	Data selection by polarizations. example: ’0’ (generate caltable for pol 0) [’0~1’,’0’] (pol 0 and 1 for first data, only 0 for second) ’’ (all polarizations)
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).
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

Parameter List

name	type	default	description
iteration	int	5	Number of iterations to perform sigma clipping to calculate threshold value of flagging.
edge	intVec	0	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. example: [10,20], [10]
flag_tsys	bool	True	Activate (True) or deactivate (False) Tsys flag.
tsys_thresh	double	3.0	Threshold value for Tsys flag.
flag_prfre	bool	True	Activate (True) or deactivate (False) flag by expected rms of pre-fit spectra.
prfre_thresh	double	3.0	Threshold value for flag by expected rms of pre-fit spectra.
flag_pofre	bool	True	Activate (True) or deactivate (False) flag by expected rms of post-fit spectra.
pofre_thresh	double	1.3333	Threshold value for flag by expected rms of post-fit spectra.
flag_prfr	bool	True	Activate (True) or deactivate (False) flag by rms of pre-fit spectra.
prfr_thresh	double	4.5	Threshold value for flag by rms of pre-fit spectra.

flag_pofr	bool	True	Activate (True) or deactivate (False) flag by rms of post-fit spectra.
pofr_thresh	double	4.0	Threshold value for flag by rms of post-fit spectra.
flag_prfrm	bool	True	Activate (True) or deactivate (False) flag by running mean of pre-fit spectra.
prfrm_thresh	double	5.5	Threshold value for flag by running mean of pre-fit spectra.
prfrm_nmean	int	5	Number of channels for running mean of pre-fit spectra.
flag_pofrm	bool	True	Activate (True) or deactivate (False) flag by running mean of post-fit spectra.
pofrm_thresh	double	5.0	Threshold value for flag by running mean of post-fit spectra.
pofrm_nmean	int	5	Number of channels for running mean of post-fit spectra.
plotflag	bool	True	True to plot result of data flagging.
pipelinemode	string	automatic	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.
infiles	stringVec	None	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.
antenna	string	None	Data selection by antenna names or ids. example: 'PM03,PM04' '' (all antennas)
field	string	None	Data selection by field names or ids. example: '*Sgr*,M100' '' (all fields)
spw	string	None	Data selection by spw ids. example: '3,4' (spw 3 and 4) '' (all spws)
pol	string	None	Data selection by polarizations. example: 'XX,YY' (correlation XX and YY) '' (all polarizations)
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).
parallel	string	automatic	Execute using CASA HPC functionality, if available.

Back

hsd_exportdata

Task Description

Prepare single dish data for export
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	Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'. Parameter is not available when pipelinemode='automatic'. example: pprfile=['PPR_GRB021004.xml']
targetimages	stringVec	None	List of science target images to be exported. Defaults to all science target images recorded in the pipeline context. Parameter is not available when pipelinemode='automatic'. example: targetimages=['r_aqr.CM02.spw5.line0.XXYY.sd.im', 'r_aqr.CM02.spw5.XXYY.sd.cont.im']
products_dir	string	None	Name of the data products subdirectory. Defaults to './' Parameter is not available when pipelinemode='automatic'. example: products_dir='../products'
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the task (False) or display task command (True). Only available when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Only available when pipelinemode='interactive'.

Back

hsd_flagdata

Task Description

Do basic flagging of a list of MeasurementSets

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	stringVec	None	The list of input MeasurementSets. Defaults to the list of MeasurementSets defined in the pipeline context.
autocorr	bool	False	Flag autocorrelation data.
shadow	bool	True	Flag shadowed antennas.
scan	bool	True	Flag a list of specified scans.
scannumber	string	None	A string containing a comma delimited list of scans to be flagged.
intents	string	POINTING,FOCUS,ATMOSPHERE,SIDEBAND,CHECK	A string containing a comma delimited list of intents against which the scans to be flagged are matched. example: '*BANDPASS*'
edgespw	bool	True	Flag the edge spectral window channels.
fracspw	any	1.875GHz	Fraction of the baseline correlator TDM edge channels to be flagged.
fracspwfps	double	0.048387	Fraction of the ACS correlator TDM edge channels to be flagged.
online	bool	True	Apply the online flags.

fileonline	string	None	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.
template	bool	True	Apply a flagging template.
filetemplate	stringVec	None	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.
pointing	bool	True	Apply a flagging template for pointing flag.
filepointing	stringVec	None	The name of a text file that contains the flagging template for pointing flag. If the template flags files is undefined a name of the form 'msname.flagpointing.txt' is assumed.
incompleteraster	bool	True	Apply commands to flag incomplete raster sequence. If this is False, relevant commands in filepointing are simply commented out.
hm_tbuff	string	halfint	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.

tbuff	any	0.0	The time in seconds used to pad flagging command time intervals if hm_tbuff='manual'.
qa0	bool	True	QA0 flags
qa2	bool	True	QA2 flags
pipelinemode	string	automatic	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.
flagbackup	bool	False	Back up any pre-existing flags before applying new ones.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hsd_imaging

Task Description

Generate single dish images

Parameter List

name	type	default	description
mode	string	line	imaging mode controls imaging parameters in the task. Accepts either "line" (spectral line imaging) or "ampcal" (image settings for amplitude calibrator)
restfreq	stringVec	None	Rest frequency
pipelinemode	string	automatic	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.
infiles	stringVec	None	List of data files. These must be a name of MeasurementSets that are registered to context via hsd_importdata task. example: vis=['uid___A002_X85c183_X36f.ms', 'uid___A002_X85c183_X60b.ms']
field	string	None	Data selection by field names or ids. example: "*Sgr*,M100"
spw	any	None	Data selection by spw ids. example: "3,4" (generate images for spw 3 and 4)
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hsd_importdata

Task Description

Imports data into the single dish pipeline

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	stringVec	None	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. example: vis=['X227.ms', 'asdms.tar.gz']
session	stringVec	None	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. example: session=['Session_1', 'Sessions_2']
pipelinemode	string	automatic	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.
asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR SpectralWindow	ASDM tables to convert as is Parameter is not available when pipelinemode='automatic'. example: 'Receiver', ''
process_caldevice	bool	False	Ingest the ASDM caldevice table. Parameter is not available when pipelinemode='automatic'. example: True

overwrite	bool	False	Overwrite existing files on import. Can only be set in pipelinemode='interactive'. When converting ASDM to MS, if overwrite=False and the MS already exists in output directory, then this existing MS dataset will be used instead.
nocopy	bool	False	Disable copying of MS to working directory Parameter is not available when pipelinemode='automatic'.
bdfflags	bool	True	Apply BDF flags on line. Parameter is not available when pipelinemode='automatic'.
lazy	bool	False	Use the lazy filter import
with_pointing_correction	bool	True	add (ASDM::Pointing::encoder - ASDM::Pointing::pointingDirection) to the value to be written in MS::Pointing::direction
createmms	string	false	Create an MMS
dryrun	bool	False	Run the task (False) or display task command (True). Parameter is available only when pipelinemode='interactive'.
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False). Parameter is available only when pipelinemode='interactive'.

Back

---

hsd_k2jycal

Task Description

Derive Kelvin to Jy calibration tables

1. Compute the Kevin to Jy calibration tables for a list of MeasurementSets:
   hsd_k2jycal()

Parameter List

name	type	default	description
dbservice	bool	True	Whether or not accessing Jy/K DB to retrieve conversion factors.
endpoint	string	asdm	Which endpoints to use for query options: 'asdm', 'model-fit', 'interpolation'
reffile	string	jyperk.csv	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. example: reffile='', reffile='working/jyperk.csv'
pipelinemode	string	automatic	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.
infiles	stringVec	None	List of input MeasurementSets. example: vis='ngc5921.ms'
caltable	stringVec	None	Name of output gain calibration tables. example: caltable='ngc5921.gcal'
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hsd_restoredata

Task Description

Restore flagged and calibration single dish data from a pipeline run

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	stringVec	None	List of raw visibility data files to be restored. Assumed to be in the directory specified by rawdata_dir. example: vis=['uid___A002_X30a93d_X43e']
session	stringVec	None	List of sessions one per visibility file. example: session=['session_3']
products_dir	string	../products	Name of the data products directory to copy calibration products from. The parameter is effective only when copytoraw = True. When copytoraw = False, calibration products in rawdata_dir will be used. example: products_dir='myproductspath'
copytoraw	bool	True	Copy calibration and flagging tables from products_dir to rawdata_dir directory. example: copytoraw=False
rawdata_dir	string	../rawdata	Name of the raw data directory. example: rawdata_dir='myrawdatapath'
lazy	bool	False	Use the lazy filler option example: lazy=True
bdfflags	bool	True	Set the BDF flags example: bdfflags=False
ocorr_mode	string	ao	Set ocorr_mode example: ocorr_mode='ca'
asis	string	SBSummary ExecBlock Antenna Station Receiver Source CalAtmosphere CalWVR	Set list of tables to import asis. example: asis='Source Receiver'
pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

hsd_skycal

Task Description

Calibrate data

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. 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. options: 'auto', 'ps', 'otf', 'otfraster'
fraction	any	10%	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. options: String style like '20%', or float value less than 1.0. For 'otfraster' mode, you can also specify 'auto'.
noff	int	-1	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'. options: any positive integer value
width	double	0.5	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. options: any float value
elongated	bool	False	Sub-parameter for calmode. Edge marking parameter for 'otf' mode. Please set True only if observed area is elongated in one direction.

pipelinemode	string	automatic	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.
infiles	stringVec	None	List of data files. These must be a name of MeasurementSets that are registered to context via hsd_importdata task. example: vis=['X227.ms', 'X228.ms']
field	string	None	Data selection by field name.
spw	any	None	Data selection by spw. (defalut all spws) example: '3,4' (generate caltable for spw 3 and 4) ['0','2'] (spw 0 for first data, 2 for second)
scan	any	None	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)
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

---

hsd_tsysflag

Task Description

Flag deviant system temperature measurements

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	stringVec	None	List of input MeasurementSets (Not used)

caltable	stringVec	None	List of input Tsys calibration tables default: [] - Use the table currently stored in the pipeline context. example: caltable=['X132.ms.tsys.s2.tbl']
flag_nmedian	bool	True	True to flag Tsys spectra with high median value.
fnm_limit	double	2.0	Flag spectra with median value higher than fnm_limit * median of this measure 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 > 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 that provides the reference shape for 'flag_fieldshape'.
ff_max_limit	double	13	Flag Tsys spectra with 'fieldshape' metric values > 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 > fb_sharps_limit.

flag_toomany	bool	True	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.
tmf1_limit	double	0.666	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.
tmef1_limit	double	0.666	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.
metric_order	string	nmedian,derivative,edgechans,fieldshape,birdies,toomany	Order in which to evaluate the flagging metrics that are enables. Disabled metrics are skipped.
normalize_tsys	bool	False	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.
filetemplate	stringVec	None	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.

pipelinemode	string	automatic	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.
dryrun	bool	False	Run the commands (True) or generate the commands to be run but do not execute (False).
acceptresults	bool	True	Add the results of the task to the pipeline context (True) or reject them (False).

Back

Created at Wed Sep 8 10:27:27 2021 UTC
This page is automatically generated from XML interface definition files.