

ALMA Science Pipeline Reference Manual: CASA 4.7.2

Interferometric and Single-Dish Data



www.almascience.org

ALMA, an international astronomy facility, is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), NSC and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ.

User Support:

For further information or to comment on this document, please contact your regional Helpdesk through the ALMA User Portal at www.almascience.org. Helpdesk tickets will be directed to the appropriate ALMA Regional Center at ESO, NAOJ or NRAO.

Revision History:

| Version | Date | Editors |
|---------------------|--------------|---------------|
| 3.14v1.0 CASA 4.5.1 | January 2016 | Pipeline Team |
| 4.14v1.0 CASA 4.7.0 | October 2016 | Pipeline Team |
| 4.14v2.0 CASA 4.7.2 | April 2017 | Pipeline Team |

In publications, please refer to this document as:

ALMA Pipeline Team, 2017, ALMA Science Pipeline Reference Manual CASA 4.7.2, ALMA Doc 4.14v2.0

Table of Contents

| | | |
|----------|---|-----------|
| 1 | PURPOSE AND SCOPE | 5 |
| 2 | PIPELINE TASK TYPES | 5 |
| 3 | PIPELINE CONTEXT | 5 |
| 4 | PIPELINE TASK LIST | 6 |
| 5 | COMMON TASK DESCRIPTIONS..... | 9 |
| 5.1 | h_init..... | 9 |
| 5.2 | h_resume..... | 10 |
| 5.3 | h_save | 11 |
| 5.4 | h_weblog | 11 |
| 6 | INTERFEROMETRY COMMON TASK DESCRIPTIONS..... | 12 |
| 6.1 | hif_antpos | 12 |
| 6.2 | hif_applycal | 14 |
| 6.3 | hif_atmflag..... | 16 |
| 6.4 | hif_bandpass | 18 |
| 6.5 | hif_bpflagchans | 21 |
| 6.6 | hif_checkproductsize..... | 24 |
| 6.7 | hif_cleanlist | 25 |
| 6.8 | hif_clean | 27 |
| 6.9 | hif_export_calstate | 31 |
| 6.10 | hif_exportdata..... | 32 |
| 6.11 | hif_findcont | 35 |
| 6.12 | hif_gaincal | 36 |
| 6.13 | hif_gainflag..... | 39 |
| 6.14 | hif_import_calstate | 42 |
| 6.15 | hif_importdata | 42 |
| 6.16 | hif_lowgainflag | 45 |
| 6.17 | hif_makecleanlist..... | 47 |
| 6.18 | hif_makeimages..... | 50 |
| 6.19 | hif_makeimlist | 52 |
| 6.20 | hif_mstransform | 56 |
| 6.21 | hif_normflux | 58 |
| 6.22 | hif_rawflagchans | 60 |
| 6.23 | hif_refant | 63 |
| 6.24 | hif_restoredata | 65 |

| | | |
|----------|--|------------|
| 6.25 | hif_setjy | 67 |
| 6.26 | hif_setmodels | 70 |
| 6.27 | hif_show_calstate | 72 |
| 6.28 | hif_tclean | 72 |
| 6.29 | hif_uvcontfit | 78 |
| 6.30 | hif_uvcontsub | 79 |
| 7 | INTERFEROMETRY ALMA TASK DESCRIPTIONS | 81 |
| 7.1 | hifa_antpos | 81 |
| 7.2 | hifa_bandpass | 83 |
| 7.3 | hifa_bpsolint | 88 |
| 7.4 | hifa_flagdata | 90 |
| 7.5 | hifa_flagtargets | 93 |
| 7.6 | hifa_fluxcalflag | 95 |
| 7.7 | hifa_fluxdb | 97 |
| 7.8 | hifa_gaincalsnr | 98 |
| 7.9 | hifa_gfluxscale | 100 |
| 7.10 | hifa_importdata | 104 |
| 7.11 | hifa_linpolcal | 106 |
| 7.12 | hifa_spwphaseup | 108 |
| 7.13 | hifa_timegaincal | 112 |
| 7.14 | hifa_tsyscal | 115 |
| 7.15 | hifa_tsysflag | 117 |
| 7.16 | hifa_wvrgcalflag | 120 |
| 7.17 | hifa_wvrgcal | 124 |
| 8 | SINGLE-DISH TASK DESCRIPTIONS | 128 |
| 8.1 | hsd_applycal | 128 |
| 8.2 | hsd_baseline | 130 |
| 8.3 | hsd_blflag | 132 |
| 8.4 | hsd_exportdata | 136 |
| 8.5 | hsd_flagdata | 138 |
| 8.6 | hsd_imaging | 141 |
| 8.7 | hsd_importdata | 142 |
| 8.8 | hsd_k2jycal | 145 |
| 8.9 | hsd_skycal | 147 |

1 Purpose and Scope

The purpose of this document is to describe the tasks available for the calibration and imaging of interferometry datasets using the ALMA Science Pipeline.

2 Pipeline Task Types

There are 4 types of Pipeline tasks. This document provides descriptions of the task types: h_, hif_, hifa_ and hsd_.

Table 1: Pipeline Task Types

| Task pre-fix | Task type | Description |
|---------------------|-----------------------------|--|
| h_ | Common tasks | Pipeline tasks used in the calibration and imaging of both interferometry and single-dish datasets |
| hif_ | Interferometry common tasks | Pipeline tasks used in the calibration and imaging of both ALMA and EVLA interferometry datasets |
| hifa_ | Interferometry ALMA tasks | Pipeline tasks used in the calibration and imaging of ALMA interferometry datasets only |
| hsd_ | Single-dish tasks | Pipeline tasks used in the calibration and imaging of single-dish datasets only |

3 Pipeline Context

The Pipeline state is stored in its context e.g. which calibration tables need to be used at each stage. Several Pipeline tasks are associated with initialising and editing (rarely needed) the context.

4 Pipeline Task List

Commissioned tasks are those used in the current ALMA standard recipes (`casa_pipescript.py` and `casa_piperestorescript.py`). Tasks which do not form part of the standard recipes are currently experimental. To assess which tasks were used in the processing of an ALMA dataset, please see the `casa_pipescript.py` and `casa_piperestorescript.py` included in ALMA deliveries for Pipeline-processed data.

Table 2: Common Tasks

| | Task Name | Description |
|---|-----------------------|---|
| 1 | <code>h_init</code> | Initialise the interferometry pipeline |
| 2 | <code>h_resume</code> | Restore a save pipeline state from disk |
| 3 | <code>h_save</code> | Save the pipeline state to disk |
| 4 | <code>h_weblog</code> | Open the pipeline weblog in a browser |

Table 3: Interferometry Common Tasks

| | Task Name | Description |
|----|----------------------|---|
| 1 | hif_antpos | Derive an antenna position calibration table |
| 2 | hif_applycal | Apply the calibration(s) to the data |
| 3 | hif_atmflag | Flag channels with bad atmospheric transmission |
| 4 | hif_bandpass | Compute bandpass calibration solutions |
| 5 | hif_bpflagchans | Flag deviant channels in bandpass calibration |
| 6 | hif_checkproductsize | Check imaging product size |
| 7 | hif_cleanlist | Compute clean map |
| 8 | hif_clean | Compute clean map |
| 9 | hif_export_calstate | Save the pipeline calibration state to disk |
| 10 | hif_exportdata | Prepare interferometry data for export |
| 11 | hif_findcont | Find continuum frequency ranges |
| 12 | hif_gaincal | Determine temporal gains from calibrator observations |
| 13 | hif_gainflag | Flag antennas with deviant gain |
| 14 | hif_import_calstate | Import a calibration state from/to disk |
| 15 | hif_importdata | Imports data into the interferometry pipeline |
| 16 | hif_lowgainflag | Flag antennas with low or high gain |
| 17 | hif_makecleanlist | Compute list of clean images to be produced |
| 18 | hif_makeimages | Compute clean map |
| 19 | hif_makeimlist | Compute list of clean images to be produced |
| 20 | hif_mstransform | Select data from calibrated MS(s) to form new MS(s) for imaging |
| 21 | hif_normflux | Average calibrator fluxes across measurement sets |
| 22 | hif_rawflagchans | Flag deviant baseline/channels in raw data |
| 23 | hif_refant | Select the best reference antennas |
| 24 | hif_restoredata | Restore flagged and calibration interferometry data from a pipeline run |
| 25 | hif_setjy | Fill the model column with calibrated visibilities |
| 26 | hif_setmodels | Set calibrator source models |
| 27 | hif_show_calstate | Show the current pipeline calibration state |
| 28 | hif_tclean | Compute clean map |
| 29 | hif_uvcontfit | Fit the continuum in the UV plane |
| 30 | hif_uvcontsub | Subtract the fitted continuum from the data |

Table 4: Interferometry ALMA Tasks

| | Task Name | Description |
|----|------------------|---|
| 1 | hifa_antpos | Derive an antenna position calibration table |
| 2 | hifa_bandpass | Compute bandpass calibration solutions |
| 3 | hifa_bpsolint | Compute optimal bandpass calibration solution intervals |
| 4 | hifa_flagdata | Do basic flagging |
| 5 | hifa_flagtargets | Do science target flagging |
| 6 | hifa_fluxcalflag | Locate and flag line regions in solar system flux calibrators |
| 7 | hifa_fluxdb | Connect to flux calibrator database |
| 8 | hifa_gaincalsnr | Compute gaincal signal to noise ratios per spw |
| 9 | hifa_gfluxscale | Derive flux density scales from standard calibrators |
| 10 | hifa_importdata | Imports data into the interferometry pipeline |
| 11 | hifa_linpolcal | Compute polarization calibration |
| 12 | hifa_spwphaseup | Compute phase calibration spw map and per spw phase offsets |
| 13 | hifa_timegaincal | Determine temporal gains from calibrator observations |
| 14 | hifa_tsyscal | Derive a Tsys calibration table |
| 15 | hifa_tsysflag | Flag deviant system temperature measurements |
| 16 | hifa_wvrgcalflag | Calculate WVR corrections |
| 17 | hifa_wvrgcal | Compute the WVR calibration |

Table 5: Single-Dish Tasks

| | Task Name | Description |
|---|------------------|---|
| 1 | hsd_applycal | Apply the calibration(s) to the data |
| 2 | hsd_baseline | Detect and validate spectral lines, subtract baseline by masking detected lines |
| 3 | hsd_blflag | Flag spectra based on predefined criteria of the single-dish pipeline |
| 4 | hsd_exportdata | Prepare singledish data for export |
| 5 | hsd_flagdata | Do basic flagging of a list of measurement sets |
| 6 | hsd_imaging | Generate single dish images |
| 7 | hsd_importdata | Import data into the single dish pipeline |
| 8 | hsd_k2jycal | Derive Kelvin to Jy calibration tables |
| 9 | hsd_skycal | Calibrate data |

5 Common Task Descriptions

5.1 h_init

h_init must be called before any other interferometry pipeline task. The pipeline can be initialised in one of two ways: by creating a new pipeline state (h_init) or by loading a saved pipeline state (h_resume). h_init creates an empty pipeline context but does not load visibility data into the context. hif_importdata or hsd_importdata can be used to load data.

Task Description

Initialise the interferometry pipeline
The h_init task initialises the interferometry pipeline.

Keyword arguments:

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

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

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

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

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

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

--- 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. Create the pipeline context
h_init()

Parameter List

Table 6: h_init default settings

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

5.2 h_resume

h_resume restores a name pipeline state from disk allowing a suspended pipeline reduction session to be resumed.

Task Description

Restore a save pipeline state from disk

h_resume restores a name pipeline state from disk allowing a suspended pipeline reduction session to be resumed.

Keyword parameters:

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

example: filename='context.s3.2012-02-13T10:49:11'
filename='last'

Examples

1. Resume the last saved session

h_resume()

2. Resume the named saved session
h_resume(filename='context.s3.2012-02-13T10:49:11')

Parameter List

Table 7: h_resume default settings

| Parameter | Type | Default | Description |
|-----------|--------|---------|--|
| filename | string | last | Filename of saved state to be restored |

5.3 h_save

h_save saves the current pipeline state to disk under a unique name. If no name is supplied one is generated automatically from a combination of the rootname 'context', the current stage number, and a timestamp.

Task Description

Save the pipeline state to disk
h_save saves the current pipeline state to disk under a unique name.

Keyword arguments:

filename -- Name of the saved pipeline state. If filename is "" then a unique name will be generated computed from the root 'context', the current stage number, and the timestamp.
default: ""

Examples

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

Parameter List

Table 8: h_save default settings

| Parameter | Type | Default | Description |
|-----------|--------|---------|----------------------|
| filename | string | None | Name for saved state |

5.4 h_weblog

Task Description

Open the pipeline weblog in a browser

Parameter List

No parameters

6 Interferometry Common Task Descriptions

6.1 hif_antpos

The hif_antpos task corrects the antenna positions recorded in the ASDMs using updated antenna position calibration information determined after the observation was taken. Corrections can be input by hand, read from a file on disk, or in future by querying an ALMA database service. The antenna positions file is in 'csv' format containing 6 comma delimited columns as shown below. The default name of this file is 'antennapos.csv'

List if sample antennapos.csv file

```
ms,antenna,xoffset,yoffset,zoffset,comment
uid___A002_X30a93d_X43e.ms,DV11,0.000,0.010,0.000,"No comment"
uid___A002_X30a93d_X43e.dup.ms,DV11,0.000,-0.010,0.000,"No comment"
```

The corrections are used to generate a calibration table which is recorded in the pipeline context and applied to the raw visibility data, on the fly to generate other calibration tables, or permanently to generate calibrated visibilities for imaging.

Task Description

Derive an antenna position calibration table

Keyword arguments:

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

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

hm_antpos -- Heuristics method for retrieving the antenna position corrections. The options are 'online' (not yet implemented), 'manual', and 'file'.
default: 'manual'
example: hm_antpos='file'

antenna -- The list of antennas for which the positions are to be corrected if hm_antpos is 'manual'
default: none
example 'DV05,DV07'

offsets -- The list of antenna offsets for each antenna in 'antennas'. Each offset is a set of 3 floating point numbers separated by commas, specified in the ITRF frame.

default: none

example: [0.01, 0.02, 0.03, 0.03, 0.02, 0.01]

antposfile -- The file(s) containing the antenna offsets. Used if hm_antpos is 'file'. The default file name is 'antennapos.csv'

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

vis -- List of input visibility files

default: []

example: ['ngc5921.ms']

caltable -- Name of output gain calibration tables

default: []

example: caltable=['ngc5921.gcal']

-- **Pipeline task execution modes**

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

default: True

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

default: True

Output

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

Issues

The hm_antpos 'online' option will be implemented when the observing system provides an antenna position determination service.

Examples

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='myantposfile.csv')

Parameter List

Table 9: hif_antpos default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|-----------|--|
| vis | stringArray | None | List of input measurement sets |
| caltable | stringArray | None | List of output caltable(s) |
| hm_antpos | string | manual | The antenna position determination method |
| antenna | string | None | List of antennas to be corrected |
| offsets | doubleArray | None | List of position corrections one set per antenna |
| antposfile | string | None | File containing antenna position corrections |
| pipelinemode | string | automatic | The pipeline operation mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically accept results into context |

6.2 hif_applycal

hif_applycal applies the precomputed calibration tables stored in the pipeline context to the set of visibility files using predetermined field and spectral window maps and default values for the interpolation schemes.

Users can interact with the pipeline calibration state using the tasks hif_export_calstate and hif_import_calstate.

Task Description

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

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

applymode -- Calibration apply mode

"='calflagstrict': calibrate data and apply flags from solutions using the strict flagging convention

'trial': report on flags from solutions, dataset entirely unchanged

'flagonly': apply flags from solutions only, data not calibrated

'calonly': calibrate data only, flags from solutions NOT applied

'calflagstrict':

'flagonlystrict': same as above except flag spws for which calibration is unavailable in one or more tables (instead of allowing them to pass uncalibrated and unflagged)

default: "

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

default: 'automatic'.

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the pipeline context.

default: []

example: ['X227.ms']

field -- A string containing the list of field names or field ids to which the calibration will be applied. Defaults to all fields in the pipeline context.

default: "

example: '3C279', '3C279, M82'

intent -- A string containing a the list of intents against which the selected fields will be matched. Defaults to all supported intents in the pipeline context.

default: "

example: '*TARGET*'

spw -- The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context.

default: "

example: '17', '11, 15'

antenna -- The list of antennas to which the calibration will be applied. Defaults to all antennas. Not currently supported.

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Issues

There is some discussion about the appropriate values of calwt. Given properly scaled data, the correct value should be the CASA default of True. However at the current time ALMA is suggesting that calwt be set to True for applying observatory calibrations, e.g. antenna positions, WVR, and system temperature corrections, and to False for applying instrument calibrations, e.g. bandpass, gain, and flux.

Examples

1. Apply the calibration to the target data

hif_applycal (intent='TARGET')

Parameter List

Table 10: hif_applycal default settings

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

6.3 hif_atmflag

Spectral window channels with low atmospheric transmission are identified and flagged. The flagging view comprises a transmission spectrum for each spectral window calculated using the CASA atmosphere model.

Flags are generated by running the following rules on each spectrum:

If flag_minabs = True then channels with transmission below fmin_limit are flagged.

If flag_nmedian = True then channels with transmission below fnm_limit * median transmission are flagged.

If flag_minabs = True then channels with transmission below fmin_limit are flagged.

If flag_nmedian = True then channels with transmission below fnm_limit * median transmission are flagged.

Task Description

Flag channels with bad atmospheric transmission

hif_atmflag flags channels where the atmospheric transmission is low

Keyword arguments:

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

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

intents -- Specifies the data intents whose channels are to be flagged if they have low atmospheric transmission. This string is inserted into the flagcmd given to the flagdata task applying the flags; it must have a valid flagcmd format.
default '*AMP*,*BANDPASS*,*PHASE*'

flag_minabs -- True to flag channels with transmission < fmin_limit.
default False

fmin_limit -- The atmospheric transmission below which channels are to be flagged if flag_minabs is True.
Default: 0.1

flag_nmedian -- True to flag channels with transmission < fmin_limit * median transmission.
default: False

fmin_limit -- Flag channels with transmission < fmin_limit * median transmission, if flag_nmedian is True.
default: 0.5

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

vis -- List of input measurement sets
default: [] - Use the measurement sets currently stored in the pipeline context.
example: vis=['X132.ms']

-- Pipeline task execution modes

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

acceptresults -- This parameter has no effect. The Tsyscal file is already in the pipeline context and is flagged in situ.

Output

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

Examples

1. Flag channels with transmission below 0.1 in each SpW.

hif_atmflag()

equivalent to:

hif_atmflag(flag_minabs=True, fmin_limit=0.1)

2. Flag channels with transmission below $0.4 * \text{median transmission across the spectral window}$.

hif_atmflag(flag_nmedian=True, fnm_limit=0.4)

Parameter List

Table 11: hif_atmflag default settings

| Parameter | Type | Default | Description |
|---------------|-------------|----------------------------------|---|
| vis | stringArray | None | List of input measurement sets |
| intent | string | *AMP*, *BANDPASS*, *PHASE* | Data intents to which flags are to be applied |
| flag_minabs | bool | False | True to flag channels where transmission < fmin_limit |
| fmin_limit | double | 0.1 | Transmission limit below which channels are to be flagged |
| flag_nmedian | bool | False | True to flag channels where transmission < fnm_limit * median transmission |
| fnm_limit | double | 0.5 | If flag_nmedian then flag channels where transmission < fnm_limit * median transmission |
| pipelinemode | string | automatic | The pipeline operations mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically apply results to context |

6.4 hif_bandpass

hif_bandpass computes a bandpass solution for every specified science spectral window. By default a 'phaseup' pre-calibration is performed and applied on the fly to the data, before the bandpass is computed. The hif_refant task may be used to precompute a prioritized list of reference antennas.

Task Description

Compute bandpass calibration solutions

Compute amplitude and phase as a function of frequency for each spectral window in each measurement set.

Previous calibration can be applied on the fly.

Keyword arguments:

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

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

phaseup -- Do a phaseup on the data before computing the bandpass solution
default: True

phaseupsolint -- The phase correction solution interval in CASA syntax. Used when phaseup is True.
default: 'int'
example: 300

phaseupbw -- Bandwidth to be used for phaseup. Defaults to 500MHz. Used when phaseup is True.
default: "
example: " default to entire bandpass, '500MHz' use centreal 500MHz

hm_bandtype -- The type of bandpass. The options are 'channel' and 'polynomial' for CASA bandpass types = 'B' and 'BPOLY' respectively.

solint -- Time and channel solution intervals in CASA syntax.
default: 'inf,7.8125MHz'
example: 'inf,10ch', 'inf'

combine -- Data axes to combine for solving. Axes are "", 'scan','spw','field' or any comma-separated combination.
default: 'scan'
example: combine='scan,field'

minblperant -- Minimum number of baselines required per antenna for each solve Antennas with fewer baselines are excluded from solutions. Used for hm_bandtype='channel' only.
default: 4

minsnr -- Solutions below this SNR are rejected. Used for hm_bandtype='channel' only
default: 3.0

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

vis -- The list of input measurement sets. Defaults to the list of measurment sets specified in the pipeline context.
default: "
example: ['M51.ms']

caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.
default: "
example: ['M51.bcal']

field -- The list of field names or field ids for which bandpasses are computed. Defaults to all fields.

default: "
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to all data with bandpass intent.

default: "
example: '*PHASE*'

spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.

default: "
example: '11,13,15,17'

refant -- Reference antenna names. Defaults to the value(s) stored in the pipeline context. If undefined in the pipeline context defaults to the CASA reference antenna naming scheme.

default: "
example: refant='DV01', refant='DV06,DV07'

solnorm -- Normalise the bandpass solutions
default: False

--- pipeline task execution modes

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

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

Output

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

Issues

There is currently some discussion about whether or not to do an 'ampup' operations at the same time as the 'phaseup'. This is not required for the bandpass computation but the amplitude information may provide a useful quality assessment measure.

The specified minsnr parameter is currently applied to the bandpass solution computation but not the 'phaseup' computation. Some noisy solutions in the phaseup may not be properly rejected.

Examples

1. Compute a channel bandpass for all visibility files in the pipeline context using the CASA reference antenna determination scheme.
hif_bandpass()

2. Same as the above but precompute a prioritized reference antenna list
hif_refant()

hif_bandpass()

Parameter List

Table 12: hif_bandpass default settings

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

6.5 hif_bpflagchans

Task Description

Flag deviant channels in bandpass calibrated data

hif_bpflagchans flags deviant channels in the bandpass corrected data.

The flagging views used derive from a bandpass calibration of PHASE data with the current context calibrations preapplied; in a normal run these would be Tsys, wvrgcal and bandpass. The 'view bandpass' results are folded into a 2d image for each spectral window with axes 'Channel' and 'Antenna ID'.

If the PHASE calibrator has a flat spectrum and the context calibration is working properly then the view bandpass results should be flat for each spectral window.

Bad channels are flagged in the underlying datasets for all intents, not just the one forming the basis of the flagging views.

Two flagging methods are available:

If parameter flag_hilo is set True then outliers from the median of each flagging view will be flagged.

If parameter flag_tmf is set True then all channels for a particular antenna/spw will be flagged if more than a specified proportion of them are already flagged for another reason.

Keyword arguments:

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

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

flag_hilo -- True to flag channel/antenna data further from the view median than fhl_limit * MAD.
default: True

fhl_limit -- If flag_hilo is True then flag channel/antenna data further from the view median than fhl_limit * MAD.
default: 7

fhl_minsample -- Do no flagging if the view median and MAD are derived from fewer than fhl_minsample view pixels.
default: 5

flag_tmf -- True to flag all channels for an antenna if the proportion of channels already flagged is greater than tmf_limit.
default: True

tmf_limit -- If flag_tmf is True then all channels will be flagged for an antenna if more than tmf_limit of them are already flagged.
default: 0.3

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

vis -- List of input measurement sets.

default: [] - Use the measurement sets currently known to the pipeline context.

-- Pipeline task execution modes

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

default: True

acceptresults -- This parameter has no effect. Any flags are applied to the measurement set data and the pipeline context is not modified.

Output

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

Examples

1. Flag birdies in the bandpass calibration for each antenna/SpW. Flag the entire calibration spectrum if more than 30 percent of channels are flagged.

hif_bpflagchans()

equivalent to:

hif_bpflagchans(flag_hilo=True, fhl_limit=7, flag_tmf=True, tmf_limit=0.3)

Parameter List

Table 13: hif_bpflagchans default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|-----------|---|
| vis | stringArray | None | List of input measurement sets |
| flag_hilo | bool | True | True to flag outlier channels |
| fhl_limit | double | 7 | Flag channels further from median than limit * MAD |
| fhl_minsample | double | 5 | Minimum number of points in sample |
| flag_tmf | bool | True | True to flag all channels if proportion of channels flagged > tmf_limit |
| tmf_limit | double | 0.3 | Fraction of channels flagged that triggers flagging of all channels |
| pipelinemode | string | automatic | The pipeline operations mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically apply results to context |

6.6 hif_checkproductsize

Task Description

Check imaging product size

Check interferometry imaging product size and try to mitigate to maximum allowed values. The task implements a mitigation cascade computing the largest cube size and trying to reduce it below a given limit by adjusting the nbins, hm_imsz and hm_cell parameters. If this step succeeds, it also checks the overall imaging product size and if necessary reduces the number of fields to be imaged.

Keyword arguments:

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

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

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the h_init or hif_importdata sets.

example: vis='ngc5921.ms'

vis=['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

default: use all measurement sets in the context

maxcubesize -- Maximum allowed cube size in GB. Used for product size mitigation.

default: -1 (automatic from performance parameters)

example: 30.0

maxproductsize -- Maximum allowed product size in GB. Used for product size mitigation.

default: -1 (automatic from performance parameters)

example: 200.0

parallel -- use multiple CPU nodes to compute dirty images

default: 'automatic'

--- pipeline task execution modes

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

default: False

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

Output

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

Parameter List

Table 14: hif_checkproductsize default settings

| Parameter | Type | Default | Description |
|-----------------------|-------------|-----------|--|
| vis | stringArray | None | List of input measurement sets |
| maxcubysize | double | -1.0 | Maximum cube size |
| maxproductsize | double | -1.0 | Maximum product size |
| parallel | string | automatic | Compute dirty images using MPI cluster |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results to the pipeline context |

6.7 hif_cleanlist

Task Description

Compute clean map
 Compute a cleaned image for a particular target source/intent and spectral window.

Keyword arguments:

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

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

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the h_init or hif_importdata sets.

example: vis='ngc5921.ms'

vis=['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

default: use all measurement sets in the context

weighting -- Weighting to apply to visibilities:

default='natural'

example: weighting='uniform'

Options: 'natural','uniform','briggs', 'superuniform','briggsabs','radial'

weighting_robust -- For weighting='briggs' and 'briggsabs'

default=0.0

example: robust=0.5

Options: -2.0 to 2.0; -2 (uniform)/+2 (natural)

weighting_noise -- For weighting='briggsabs' noise parameter to use for Briggs "abs" weighting

example: noise='1.0mJy'

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Parameter List

Table 15: hif_cleanlist default settings

| Parameter | Type | Default | Description |
|--------------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets |
| target_list | any | {} | Dictionary specifying targets to be imaged; blank will read list from context |
| weighting | string | natural | Weighting of uv (natural, uniform, briggs, ...) |
| robust | double | 0.0 | Briggs robustness parameter |
| noise | any | 1.0Jy | Noise parameter for Briggs abs mode weighting |
| npixels | int | 1 | Number of pixels for superuniform or Briggs weighting |
| hm_masking | string | None | Pipeline heuristics masking option |

| | | | |
|----------------------|--------|-----------|--|
| hm_cleaning | string | None | Pipeline cleaning mode |
| tlimit | double | 2.0 | Times the sensitivity limit for cleaning |
| masklimit | int | 4 | Times good mask pixels for cleaning |
| maxncleans | int | 1 | Maximum number of clean task calls |
| 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 |

6.8 hif_clean

Task Description

Compute clean map

Compute a cleaned image for a particular target source/intent and spectral window.

Keyword arguments:

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

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

default: 'automatic'.

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the context.

default: ""

example: vis=['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

imagename -- Prefix of output images. Defaults to one of the following options depending on the availability of project information.

'{ousstatus uid}.{field}.[{intent}.]s{stage number}.spw{spw}'

'multivis.{field}.[{intent}.]s{stage number}.spw{spw}'

Cleanboxes and thresholds to use as it goes. For each iteration the output images are:

{prename}.iter{n}.image; cleaned and restored image

{prename}.iter{n}.psf; point spread function (dirty beam)

{prename}.iter{n}.flux; relative sky sensitivity over field

{prename}.iter{n}.flux.pbcoverage; relative pb coverage over field

(only for mosaics)

{prename}.iter{n}.model; image of clean components

{prename}.iter{n}.residual; image of residuals

{prename}.iter{n}.cleanmask; image of cleanmask used

default: "
example: 'test1'

intent -- An intent against which the selected fields are matched. Default means select all data from fields specified by 'field' parameter
default: "
example: ", 'TARGET'

field -- Fields id(s) or name(s) to image or mosaic. Must be set.
default:
example: '3C279', 'Centaurus*'

spw -- Spectral window/channels to image. '\\' for all science data.
default: "
example: '9', '9,11'

spwsel -- Spectral window frequency selection. '\\' for all science data.
default: "
example: '89.1~89.5GHz'

mode -- Frequency imaging mode, 'mfs', 'frequency'. '\\' defaults to 'frequency' if intent parameter includes 'TARGET' otherwise 'mfs'.
default: "
example: 'mfs', 'frequency'

imagermode -- Advanced imaging mode e.g. mosaic or Cotton-Schwab clean.
Derived as follows:
1. The 'field' parameter is converted into a list of field_ids for each measurement set in 'vis'.
2. If there is more than 1 field_id in the list for any measurement set then imagermode is set to 'mosaic', otherwise it will be set to 'csclean'.
default: "

outframe -- The reference frame of the output image. The only supported option is 'LSRK'
default: "
example: 'LSRK'

imsize -- X and Y image size in pixels). Must be even and contain factors 2,3,5,7 only.
Default derived as follows:
1. Determine 'phasecenter' value and spread of field centres around it.
2. Set size of image to cover spread of field centres plus a border of width 0.75 * beam radius (to first null).
3. Divide x and y extents by 'cell' values to arrive at the numbers of pixels required.
default: "
example: [320,320]

cell -- X and Y cell size. Derived from maximum UV spacing. Details TBD
default "
example: ['0.5arcsec', '0.5arcsec']

phasecenter -- Direction measure or field id for the mosaic center.
Default derived as follows:
1. Make an array containing all the field centers to be imaged together.

2. Derive the mean direction from the directions array.

default: ''

example: 2

nchan -- Number of channels or planes in the output image, -1 for all

default: -1

example: 128

width -- Width of spectral dimension in frequency, '' for default.

default: ''

example: '7.8125MHz'

weighting -- Weighting to apply to visibilities. Options are: 'natural', 'uniform', 'briggs',

'superuniform', 'briggsabs', 'radial'

default='natural'

example: weighting='uniform'

robust -- Parameter for 'briggs' and 'briggsabs' weighting. Ranges from -2.0 to 2.0. -2 for uniform
+2 for natural.

default=0.0

example: 0.5

noise -- Parameter for 'briggsabs' weighting

default: '1.0Jy'

example: '0.5Jy'

npixels -- Parameter for 'briggs' and 'briggsabs; weighting

default: 1

example: 1

restoringbeam -- Gaussian restoring beam for clean, '' for default

default: ''

hm_masking -- Clean masking mode. Options are 'none', 'centralquarter' 'psf', 'psfiter' and 'manual'

default: 'centralquarter'

example: 'manual'

mask -- Image mask for hm_masking manual mode. User responsible for matching image sizes,
coordinates, etc.

default: ''

example: 'mymask.mask'

niter -- Maximum number of iterations per clean call

default: 500

example: 100

threshold -- Threshold for cleaning

default: '0.0'

example: '0.05'

maxncleans -- Maximum number of clean calls

default: 1

example: 10

--- pipeline task execution modes

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

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

Output

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

Examples

Make an 'mfs' image of calibrator 3c279 using data in spectral window 1. The cell size is set to 0.2 arcsec in RA and Dec. Other clean parameters are derived from heuristics:
`hif_clean(field='3c279', cell='0.2arcsec', spw='1', mode='mfs')`

Make a cube of calibrator 3c279 using data in spectral window 1. The cube planes will be evenly spaced in frequency in the LSRK frame. Other clean parameters are derived from heuristics.
`hif_clean(field='3c279', cell='0.2arcsec', spw='1', mode='frequency', outframe='LSRK')`

Parameter List

Table 16: hif_clean default settings

| Parameter | Type | Default | Description |
|-------------------|-------------|---------|--|
| vis | stringArray | None | List of input measurement sets, '\\' for default |
| imagename | string | None | Prefix for image filenames, '\\' for default |
| intent | string | None | Set of data selection intents, '\\' for all |
| field | string | None | Set of data selection field names or ids |
| spw | string | None | Set of data selection spectral window/channels '\\' for all |
| spwsel | string | None | Set of spw frequency selections '\\' for all |
| uvrange | any | None | Set of uv ranges, '\\' for all |
| mode | string | None | Spectral gridding type (mfs, frequency, '\\' for default) |
| imagermode | string | None | Imaging mode (csclean, mosaic, '\\' for default) |
| outframe | string | None | Velocity frame of output image (LSRK, '\\' for default) |
| imsize | intArray | None | X and Y image size in pixels, single value same for both, '\\' for default |

| | | | |
|----------------------|-------------|-----------|--|
| cell | stringArray | None | X and Y cell size(s), single value same for both, '\\' for default |
| phasecenter | any | None | Image center (direction or field index), '\\' for default |
| nchan | int | -1 | Number of channels or planes in output image, -1 = all |
| start | any | None | Start of output spectral dimension |
| width | any | None | Width of output spectral channels, '\\' for default |
| weighting | string | natural | Type of weighting |
| robust | double | 0.0 | Briggs weighting robustness parameter |
| noise | any | 1.0Jy | Briggs weighting noise parameter |
| npixels | int | 1 | Weighting algorithm parameter |
| restoringbeam | stringArray | None | Gaussian restoring beam, '\\' for default |
| hm_masking | string | none | Pipeline heuristics masking option |
| hm_cleaning | string | manual | Pipeline clean control heuristics |
| mask | any | None | User mask, '\\' for whole image |
| niter | int | 500 | Maximum number of clean iterations |
| threshold | double | 0.0 | Flux level to stop cleaning, must include units: '\\1.0mJy' |
| tlimit | double | 2.0 | Times the sensitivity limit for cleaning |
| masklimit | int | 4 | Times good mask pixels for cleaning |
| maxncleans | int | 1 | Maximum number of clean task calls |
| 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 |

6.9 hif_export_calstate

hif_export_calstate saves the current pipeline calibration state to disk in the form of a set of equivalent applycal calls. If filename is not given, hif_export_calstate saves the calibration state to disk with a filename based on the pipeline context creation time, using the extension '.calstate'

One of two calibration states can be exported: either the active calibration state (those calibrations currently applied on-the-fly but scheduled for permanent application to the measurement set in a subsequent hif_applycal call) or the applied calibration state (calibrations that were previously applied to the measurement set using hif_applycal). The default is to export the active calibration state.

Task Description

Save the pipeline calibration state to disk

`hif_export_calstate` saves the current pipeline calibration state to disk in the form of a set of equivalent `applycal` calls.

Keyword arguments:

filename -- Name for the saved calibration state.

state -- Calibration state to export.

Issues

If run several times in one pipeline session does the automatic export file naming scheme, overwrite previous versions?

Examples

1. Save the calibration state.

```
hif_export_calstate()
```

2. Save the active calibration state with a custom filename

```
hif_export_calstate(filename='afterbandpass.calstate')
```

3. Save the applied calibration state with a custom filename

```
hif_export_calstate(filename='applied.calstate', state='applied')
```

Parameter List

Table 17: `hif_export_calstate` default settings

| Parameter | Type | Default | Description |
|-----------------|--------|---------|----------------------------------|
| filename | string | None | Name for saved calibration state |
| state | string | active | The calibration state to export |

6.10 `hif_exportdata`

The `hif_exportdata` task exports the data defined in the pipeline context and exports it to the data products directory, converting and or packing it as necessary. The current version of the task exports the following products:

- an XML file containing the pipeline processing request
- a tar file per ASDM / MS containing the final flags version
- a text file per ASDM / MS containing the final calibration apply list
- a FITS image for each selected calibrator source image
- a FITS image for each selected science target source image
- a tar file per session containing the caltables for that session

- a tar file containing the file web log
- a text file containing the final list of CASA commands

Task Description

Prepare interferometry data for export

The `hif_exportdata` task exports the data defined in the pipeline context and exports it to the data products directory, converting and or packing it as necessary.

Keyword arguments:

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

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

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

vis -- List of visibility data files for which flagging and calibration information will be exported. Defaults to the list maintained in the pipeline context.
default: []
example: vis=['X227.ms', 'X228.ms']

session -- List of sessions one per visibility file. Currently defaults to a single virtual session containing all the visibility files in vis. In future will default to set of observing sessions defined in the context.
default: []
example: session=['session1', 'session2']

pprfile -- Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'.
default: []
example: pprfile=['PPR_GRB021004.xml']

calintents -- List of calibrator image types to be exported. Defaults to all standard calibrator intents 'BANDPASS', 'PHASE', 'FLUX'
default: ""
example: calintents='PHASE'

calimages -- List of calibrator images to be exported. Defaults to all calibrator images recorded in the pipeline context.
default: []
example: calimages=['3C454.3.bandpass', '3C279.phase']

targetimages -- List of science target images to be exported. Defaults to all science target images recorded in the pipeline context.
default: []

example: targetimages=['NGC3256.band3', 'NGC3256.band6']

products_dir -- Name of the data products subdirectory. Defaults to './'

default: ''

example: products_dir='./products'

--- pipeline task execution modes

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

default: True

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

default: True

Output

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

Issues

Support for merging the calibration state information into the pipeline context / results structure and retrieving it still needs to be added.

Support for merging the clean results into the pipeline context / results structure and retrieving it still needs to be added.

Support for creating the final pipeline results entity still needs to be added.

Session information is not currently handled by the pipeline context. By default all ASDMs are combined into one session.

Examples

1. Export the pipeline results for a single sessions 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

Table 18: hif_exportdata default settings

| Parameter | Type | Default | Description |
|-----------|-------------|---------|-------------------------------|
| vis | stringArray | None | List of input visibility data |

| | | | |
|----------------------|-------------|-----------|---|
| session | stringArray | None | List of sessions one per visibility file |
| pprfile | string | None | The pipeline processing request file to be exported |
| calintents | string | None | The calibrator source target intents to be exported |
| calimages | stringArray | None | List of calibrator images to be exported |
| targetimages | stringArray | None | List of target images to be exported |
| products_dir | string | None | The data products directory |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

6.11 hif_findcont

Task Description

Find continuum frequency ranges
 Compute continuum ranges for all sources and spectral windows.

Keyword arguments:

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

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

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the h_init or hif_importdata sets.

example: vis='ngc5921.ms'
 vis=['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']
 default: use all measurement sets in the context

parallel -- use multiple CPU nodes to compute dirty images
 default: 'automatic'

--- pipeline task execution modes

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

default: False

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

Output

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

Parameter List

Table 19: hif_findcont default settings

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

6.12 hif_gaincal

The complex gains are derived from the data column (raw data) divided by the model column (usually set with hif_setjy). The gains are obtained for a specified solution interval, spw combination and field combination. Good candidate reference antennas can be determined using the hif_refant task. Previous calibrations that have been stored in the pipeline context are applied on the fly. Users can interact with these calibrations via the hif_export_calstate and hif_import_calstate tasks.

Task Description

Determine temporal gains from calibrator observations
Compute the gain solutions

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

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

hm_gtype -- The type of gain calibration. The options are 'gtype' and 'gspline' for CASA gain types = 'G' and 'GSPLINE' respectively.

calmode -- Type of solution, The options are 'ap' (amp and phase), 'p' (phase only) and 'a' (amp only)
default: 'ap'
options: 'p','a','ap'

solint -- Time solution intervals in CASA syntax. Works for hm_gtype='gtype' only.
default: 'inf'
example: 'inf', 'int', '100sec'

combine -- Data axes to combine for solving. Options are "", 'scan', 'spw', 'field' or any comma-separated combination. Works for hm_gtype='gtype' only.
default: ""
example: combine=""

minblperant -- Minimum number of baselines required per antenna for each solve Antennas with fewer baselines are excluded from solutions. Works for hm_gtype='gtype' only.
default: 4
example: minblperant=2

minsnr -- Solutions below this SNR are rejected. Works for hm_gtype='channel' only
default: 3.0

splintime -- Spline timescale (sec). Used for hm_gtype='gspline'. Typical splintime should cover about 3 to 5 calibrator scans.
default: 3600 (1 hour)
example: splintime=1000

npointaver -- Tune phase-unwrapping algorithm. Used for hm_gtype='gspline'
default: 3 (Keep at this value)

phaseswrap -- Wrap the phase for changes larger than this amount (degrees) Used for hm_gtype='gspline'.
default: 180 (Keep at this value)

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context
default: ""
example: ['M82A.ms', 'M82B.ms']

caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.
default: ""
example: ['M82.gcal', 'M82B.gcal']

field -- The list of field names or field ids for which gain solutions are to be computed. Defaults to all fields with the standard intent.
default: ""
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to *PHASE*.
default: "
example: ", '*AMP*', *PHASE*

spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all science spectral windows.
default: "
example: '3C279', '3C279, M82'

smodel -- Point source Stokes parameters for source model (experimental) Defaults to using standard MODEL_DATA column data.
default: []
example: [1,0,0,0] (I=1, unpolarized)

refant -- Reference antenna name(s) in priority order. Defaults to most recent values set in the pipeline context. If no reference antenna is defined in the pipeline context use the CASA defaults.
default: "
example: refant='DV01', refant='DV05,DV07'

solnorm -- Normalise the gain solutions
default: False

--- pipeline task execution modes

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

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

Output

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

Issues

The 'gspline' (smooth) option is still under development in CASA.

Examples

1. Compute standard per scan gain solutions that will be used to calibrate the target.
hif_gaincal()

Parameter List

Table 20: hif_gaincal default settings

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

6.13 hif_gainflag

Deviant antennas are detected by analysis of a view showing their calibration gains. Three flagging metrics are available (mediandeviant, rmsdeviant, nrmsdeviant) and a flagging view is created for each metric that has been enabled. Each view is a list of 2D images with axes 'Time' and 'Antenna'; there is one image for each spectral window and intent. If any of the flagging metrics exceeds their corresponding limit (fmeddev_limit, frmsdev_limit, fnrmsdev_limit) for a given antenna in a given spw, then all data for that antenna and for all spws within the same baseband will be flagged with a flagcmd.

The following heuristics are used for the flagging metrics:

mediandeviant: $\text{abs}(\text{median}(\text{antenna}) - \text{median}(\text{all antennas})) / \text{MAD}(\text{all antennas}) > \text{fmeddev_limit}$

rmsdeviant: $\text{stdev}(\text{antenna}) / \text{MAD}(\text{all antennas}) > \text{frmsdev_limit}$

nrmsdeviant: $\text{deviation}(\text{ant_i}) = (\text{sm}(\text{ant_i}) - \text{med_sm_allant}) / \text{sigma_sm_allant}$

with: $\text{sm}(\text{ant_i}) = \text{sigma}(\text{ant_i}) / \text{median}(\text{ant_i})$

$\text{sigma_sm_allant} = 1.4826 * \text{mad}(\{\text{sm}(\text{ant_1}), \text{sm}(\text{ant_2}), \dots, \text{sm}(\text{ant_nants})\})$

$\text{med_sm_allant} = \text{median}(\{\text{sm}(\text{ant_1}), \text{sm}(\text{ant_2}), \dots, \text{sm}(\text{ant_nants})\})$

where MAD is the median absolute deviation from the median.

Task Description

Flag antennas with deviant gain

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

Keyword arguments:

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

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

flag_mediandeviant -- True to flag antennas with deviant median gain, calculated as:
 $\text{abs}(\text{median}(\text{antenna}) - \text{median}(\text{all antennas})) / \text{MAD}(\text{all antennas}) > \text{fmeddev_limit}$
default: False

fmeddev_limit -- Flag antennas with 'mediandeviant' metric larger than fmeddev_limit.
default: 3.0

flag_rmsdeviant -- True to flag antennas with deviant gain rms, calculated as:
 $\text{stdev}(\text{antenna}) / \text{MAD}(\text{all antennas}) > \text{frmsdev_limit}$
default: False

frmsdev_limit -- Flag antennas with 'rmsdeviant' metric larger than frmsdev_limit.
default: 8.0

flag_nrmsdeviant -- True to flag antennas with deviant normalised gain rms, calculated as:
 $\text{deviation}(\text{ant_i}) = (\text{sm}(\text{ant_i}) - \text{med_sm_allant}) / \text{sigma_sm_allant}$
where: $\text{sm}(\text{ant_i}) = \text{sigma}(\text{ant_i}) / \text{median}(\text{ant_i})$
 $\text{sigma_sm_allant} = 1.4826 * \text{mad}(\{\text{sm}(\text{ant_1}), \text{sm}(\text{ant_2}), \dots, \text{sm}(\text{ant_nants})\})$
 $\text{med_sm_allant} = \text{median}(\{\text{sm}(\text{ant_1}), \text{sm}(\text{ant_2}), \dots, \text{sm}(\text{ant_nants})\})$
default: True

fnrmsdev_limit -- Flag antennas with 'nrmsdeviant' metric larger than fnrmsdev_limit.
default: 6.0

metric_order -- Order in which the flagging metrics are evaluated.
default: 'mediandeviant, rmsdeviant, nrmsdeviant'

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

vis -- List of input measurement sets.
 default: [] - Use the measurement sets currently known to the pipeline context.

intent -- A string containing the list of intents to be checked for antennas with deviant gains. The default is blank, which causes the task to select the 'BANDPASS' intent.
 default: ""
 example: "*BANDPASS*"

spw -- The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context.
 default: ""
 example: '17', '11, 15'

refant -- A string containing a prioritized list of reference antenna name(s) to be used to produce the gain table. Defaults to the value(s) stored in the pipeline context. If undefined in the pipeline context defaults to the CASA reference antenna naming scheme.
 default: ""
 example: refant='DV01', refant='DV06,DV07'

-- Pipeline task execution modes

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

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

Output

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

Parameter List

Table 21: hif_gainflag default settings

| Parameter | Type | Default | Description |
|---------------------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets |
| intent | string | None | Data intent whose gains are to be checked |
| spw | string | None | Spectral window ids whose gains are to be checked |
| refant | string | None | Reference antenna names |
| flag_mediandeviant | bool | False | True to flag antennas with deviant median gains |
| fmeddev_limit | double | 3.0 | Flag antennas with '\mediandeviant\' metric above fmeddev_limit |

| | | | |
|-------------------------|--------|--|--|
| flag_rmsdeviant | bool | False | True to flag antennas with deviant gain rms |
| frmsdev_limit | double | 8.0 | Flag antennas with '\rmsdeviant\' metric above frmsdev_limit |
| flag_nrmsdeviant | bool | True | True to flag antennas with deviant normalised gain rms |
| fnrmsdev_limit | double | 6.0 | Flag antennas with '\nrmsdeviant\' metric above fnrmsdev_limit |
| metric_order | string | mediandeviant, rmsdeviant, nrmsdeviant | Order in which the flagging metrics are evaluated |
| pipelinemode | string | automatic | The pipeline operations mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically apply results to context |

6.14 hif_import_calstate

hif_import_calstate clears and then recreates the pipeline calibration state based on the set of applycal calls given in the named file. The applycal statements are interpreted in additive fashion; for identically specified data selection targets, caltables specified in later statements will be added to the state created by earlier calls.

Task Description

Import a calibration state from disk
 Import a calibration state to disk.

Keyword arguments:

filename -- Name of the saved calibration state.

Example

1. Import a calibration state from disk.
 hif_import_calstate(filename='aftergaincal.calstate')

Parameter List

Table 22: hif_import_calstate default settings

| Parameter | Type | Default | Description |
|-----------------|--------|---------|-------------------------------------|
| filename | string | None | Name of the saved calibration state |

6.15 hif_importdata

Task Description

Imports data into the interferometry pipeline

The `hif_importdata` task loads the specified visibility data into the pipeline context unpacking and / or converting it as necessary.

Keyword arguments:

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

vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs, MSs, or tar files of MSs. If ASDM files are specified, they will be converted to MS format.

default: []

example: `vis=['X227.ms', 'asdms.tar.gz']`

session -- List of sessions to which the visibility files belong. Defaults to a single session containing all the visibility files, otherwise a session must be assigned to each vis file.

default: []

example: `session=['session_1', 'session_2']`

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

default: 'automatic'.

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

asis -- ASDM tables to convert as is

default: 'Antenna Station Receiver CalAtmosphere'

example: 'Receiver', "

process_caldevice -- Ingest the ASDM caldevice table

default: False

example: True

overwrite -- Overwrite existing MSs on output.

default: False

bdfflags -- Apply BDF flags on import

default: True

lazy -- Use the lazy import option

default: False

dbservice -- Use online flux catalog on import

default: False

createmms -- Create a multi-measurement set ('true') ready for parallel processing, or a standard measurement set ('false'). The default setting ('automatic') creates an MMS if running in a cluster environment.

default: automatic

clearcals -- Clear pipeline calibrations as needed by resetting the corrected data column to the data column and setting the model column to unity. For normal pipeline operations clearcals should be left on, and the pipeline will choose when it needs to reset the calibrations. In some reprocessing applications clearcals should be turned off.

default: True

--- pipeline task execution modes

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

default: True

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

default: True

Output

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

Examples

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

```
hif_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e',  
 '../rawdata/uid_A002_x30a93d_X44e'])
```

2. Load an MS in the current directory into the context.

```
hif_importdata (vis=[uid___A002_X30a93d_X43e.ms])
```

3. Load a tarred ASDM in ../rawdata into the context.

```
hif_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']
```

```
hif_importdata(vis=myvislist, pipelinemode='getinputs')
```

```
hif_importdata(vis=myvislist)
```

5. Load an ASDM but check the results before accepting them into the context.

```
results = hif_importdata (vis=['uid___A002_X30a93d_X43e.ms'], acceptresults=False)
```

```
results.accept()
```

6. Run in dryrun mode before running for real

```
results = hif_importdata (vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)
```

```
results = hif_importdata (vis=['uid___A002_X30a93d_X43e.ms'])
```

Parameter List

Table 23: hif_importdata default settings

| Parameter | Type | Default | Description |
|--------------------------|-------------|-----------|---|
| vis | stringArray | None | List of input visibility data |
| session | stringArray | None | List of visibility data sessions |
| pipelinemode | string | automatic | The pipeline operating mode |
| asis | string | None | ASDM to convert as is |
| process_caldevice | bool | False | Import the caldevice table from the ASDM |
| overwrite | bool | False | Overwrite existing files on import |
| bdf flags | bool | True | Apply BDF flags on import |
| lazy | bool | False | Use the lazy import option |
| dbservice | bool | True | Use the online flux catalog |
| createmms | string | automatic | Create a MMS |
| clearcals | bool | True | Reinitialize pipeline calibrations as needed |
| dryrun | bool | False | Run the task (False) or display task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

6.16 hif_lowgainflag

Deviant antennas are detected by analysis of a view showing their calibration gains. This view is a list of 2D images with axes 'Time' and 'Antenna'; there is one image for each spectral window and intent. A flagcmd to flag all data for an antenna will be generated by any gain that is outside the range [f_{nm_lo_limit} * median, f_{nm_hi_limit} * median].

Task Description

Flag antennas with low or high gain
hif_lowgainflag flags data for antennas with unusually low or high gains.

Keyword arguments:

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

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

flag_nmedian -- True to flag figures of merit greater than f_{nm_hi_limit} * median or lower than f_{nm_lo_limit} * median.

default True

fnm_lo_limit -- Points lower than `fnm_lo_limit * median` are flagged.
default 0.7

fnm_hi_limit -- Points greater than `fnm_hi_limit * median` are flagged.
default 1.3

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

vis -- List of input measurement sets.
default: [] - Use the measurement sets currently known to the pipeline context.

intent -- A string containing the list of intents to be checked for antennas with deviant gains. The default is blank, which causes the task to select the 'BANDPASS' intent.
default: ""
example: "*BANDPASS*"

spw -- The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context.
default: ""
example: '17', '11, 15'

refant -- A string containing a prioritized list of reference antenna name(s) to be used to produce the gain table. Defaults to the value(s) stored in the pipeline context. If undefined in the pipeline context defaults to the CASA reference antenna naming scheme.
default: ""
example: `refant='DV01', refant='DV06,DV07'`

-- Pipeline task execution modes

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

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

Output

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

Parameter List

Table 24: hif_lowgainflag default settings

| Parameter | Type | Default | Description |
|---------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets |
| intent | string | None | Data intent whose gains are to be checked |

| | | | |
|--------------------------------|--------|-----------|---|
| spw | string | None | Spectral window ids whose gains are to be checked |
| refant | string | None | Reference antenna names |
| flag_nmedian | bool | True | True to flag values outside range [f _{nm_lo_limit} * median, f _{nm_hi_limit} * nmedian] |
| f_{nm_lo_limit} | double | 0.7 | Flag values lower than f _{nm_lo_limit} * median |
| f_{nm_hi_limit} | double | 1.3 | Flag values higher than f _{nm_hi_limit} * median |
| pipelinemode | string | automatic | The pipeline operations mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically apply results to context |

6.17 hif_makecleanlist

Generate a list of images to be cleaned. By default the list will include one image per science target per spw. Calibrator targets can be selected by setting appropriate values for itent. By default the output image cellsize is set to the minimum cell size consistent with the UV coverage. By default the image size in pixels is set to values determined by the cell size and the single dish beam size. If a calibrator is being imaged (intents 'PHASE', 'BANDPASS', 'FLUX' or 'AMPLITUDE') then the image dimensions are limited to 'calmaxpix' pixels. By default science target images are cubes and calibrator target images are single channel. Science target images may be mosaics or single fields.

Task Description

Compute list of clean images to be produced
 Create a list of images to be cleaned.

Keyword Arguments

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

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

mode -- Frequency specification:

default: 'mfs'

example:

mode='mfs' produce one image from all specified data

mode='frequency', channels are specified in frequency

mode='velocity', channels are specified in velocity

imsize -- Image X and Y size in pixels. The sizes must be even and divisible by 2,3,5,7 only.

default: " The default values are derived as follows:

1. Determine phase center and spread of field centers around it.

2. Set the size of the image to cover the spread of field centers plus a border of width $0.75 * \text{beam radius}$, to first null.

3. Divide X and Y extents by cell size to arrive at the number of pixels required.

example: [120, 120]

example: 300

calmaxpix -- Maximum image X or Y size in pixels if a calibrator is being imaged ('PHASE', 'BANDPASS', 'AMPLITUDE' or 'FLUX').

default: 300

example: 300

width -- Output channel width.

default: " Difference in frequency between first 2 selected channels for frequency mode images.

example: '24.2kHz'

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the h_init or hif_importdata sets.

default ": use all measurement sets in the context

example: 'ngc5921.ms', ['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

intent -- Select intents for which associated fields will be imaged.

default: 'TARGET'

example: 'PHASE,BANDPASS'

field -- Select fields to image. Use field name(s) NOT id(s). Mosaics are assumed to have common source / field names. If intent is specified only fields with data matching the intent will be selected. The fields will be selected from measurement sets in 'vis'.

default: " Fields matching matching intent, one image per target source.

example: '3C279', 'Centaurus*', '3C279,J1427-421'

spw -- Select spectral window/channels to image.

default: " Individual images will be computed for all science spectral windows.

example: '9'

linesfile -- Name of file with line regions to exclude for continuum images.

default: 'lines.dat'

example: 'mylines.dat'

uvrange -- Select a set of uv ranges ro image.

default: " All uv data is included

example: '0~1000klambda', ['0~100klambda', '100~1000klambda']

phasecenter -- Direction measure or field id of the image center.

default: " The default phase center is set to the mean of the field directions of all fields that are to be image together.

example: 0, 'J2000 19h30m00 -40d00m00'

nchan -- Total number of channels in the output image(s)

default: -1 Selects enough channels to cover the data selected by spw consistent with start and width.

example: 100

start -- First channel for frequency mode images.
default " Starts at first input channel of the spw.
example: '22.3GHz'

--- pipeline task execution modes

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

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

Output

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

Examples

1. Make a list of science target images to be cleaned, one image per science spw.
hif_makecleanlist()
2. Make a list of PHASE and BANDPASS calibrator targets to be imaged, one image per science spw.
hif_makecleanlist(intent='PHASE,BANDPASS')
3. Make a list of PHASE calibrator images observed in spw 1, images limited to 50 pixels on a side.
hif_makecleanlist(intent='PHASE',spw='1',calmaxpix=50)

Parameter List

Table 25: hif_makecleanlist default settings

| Parameter | Type | Default | Description |
|------------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets |
| imagename | string | None | Prefix for output image names, '\\' for default. |
| intent | string | TARGET | Set of data selection intents |
| field | string | None | Set of data selection field names or ids, '\\' for all |
| spw | string | None | Set of data selection spectral window/channels, '\\' for all |
| linesfile | string | None | Name of file with line regions to exclude for continuum images. |
| uvrange | string | None | Set of data selection uv ranges, '\\' for all. |
| mode | string | None | Spectral gridding type (mfs, frequency, '\\' for default) |

| | | | |
|----------------------|-------------|-----------|--|
| outframe | string | None | Velocity frame of output image (LSRK, '\') for default) |
| imsize | intArray | None | Image X and Y size(s) in pixels, '\') for default. Single value same for both. |
| cell | stringArray | None | Image X and Y cell size(s), '\') for default. Single value same for both |
| calmaxpix | int | 300 | Maximum X and Y size of calibrator images in pixels |
| phasecenter | any | None | Image center (direction or field index, '\') for default) |
| nchan | int | -1 | Number of channels, -1 = all |
| start | any | None | Channel start, '\') for default |
| width | any | None | Channel width, '\') for default. |
| 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 |

6.18 hif_makeimages

Task Description

Compute clean map
 Compute a cleaned image for a particular target source/intent and spectral window.

Keyword arguments:

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

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

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the h_init or hif_importdata sets.
 example: vis='ngc5921.ms'
 vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
 default: use all measurement sets in the context

weighting -- Weighting to apply to visibilities
 default='briggs'
 example: weighting='uniform'

Options: 'natural','uniform','briggs', 'superuniform','briggsabs','radial'

weighting_robust -- For weighting='briggs' and 'briggsabs'
default=-999.0
example: robust=0.7
Options: -2.0 to 2.0; -2 (uniform)/+2 (natural)

weighting_noise -- For weighting='briggsabs' noise parameter to use for Briggs "abs" weighting
example noise='1.0mJy'

hm_masking -- Clean masking mode. Options are 'centralregion', 'psf', 'psfiter', 'auto', 'auto-thresh', 'manual' and 'none'
default: 'centralregion'
example: 'manual'

hm_maskthreshold -- Auto-boxing mask threshold. See tclean for options.

cleancontranges -- Clean continuum frequency ranges in cubes
default=False
options: False, True

subcontms -- Subtract continuum in MS
Caveat: if enabled, repeated continuum subtraction will lead to erroneous negative fluxes.
default=False
Options: False, True

parallel -- use multiple CPU nodes to clean images
default: 'automatic'

--- pipeline task execution modes

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

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

Output

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

Parameter List

Table 26: hif_makeimages default settings

| Parameter | type | default | description |
|-------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets |
| target_list | any | {} | Dictionary specifying targets to be imaged; blank will read list from context |

| | | | |
|-------------------------|--------|---------------|---|
| weighting | string | briggs | Weighting of uv (natural, uniform, briggs, ...) |
| robust | double | -999.0 | Briggs robustness parameter |
| noise | any | 1.0Jy | Noise parameter for briggs abs mode weighting |
| npixels | int | 0 | Number of pixels for superuniform or Briggs weighting |
| hm_masking | string | centralregion | Pipeline heuristics masking option |
| hm_maskthreshold | string | None | Auto-boxing mask threshold |
| hm_cleaning | string | None | Pipeline cleaning mode |
| tlimit | double | 4.0 | Times the sensitivity limit for cleaning |
| masklimit | int | 4 | Times good mask pixels for cleaning |
| maxncleans | int | 1 | Maximum number of clean task calls |
| cleanconranges | bool | False | Clean continuum frequency ranges in cubes |
| subcontms | bool | False | Subtract continuum in MS |
| parallel | string | automatic | Clean images using MPI cluster |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results to the pipeline context |

6.19 hif_makeimlist

Generate a list of images to be cleaned. By default the list will include one image per science target per spw. Calibrator targets can be selected by setting appropriate values for itent. By default the output image cellsize is set to the minimum cell size consistent with the UV coverage. By default the image size in pixels is set to values determined by the cell size and the single dish beam size. If a calibrator is being imaged (intents 'PHASE', 'BANDPASS', 'FLUX' or 'AMPLITUDE') then the image dimensions are limited to 'calmaxpix' pixels. By default science target images are cubes and calibrator target images are single channel. Science target images may be mosaics or single fields.

Task Description

Compute list of clean images to be produced
 Create a list of images to be cleaned.

Keyword Arguments

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

default: 'automatic'.

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

specmode -- Frequency imaging mode, 'mfs', 'cont', 'cube'. '\ ' defaults to 'cube' if intent parameter includes 'TARGET' otherwise 'mfs'.

default: "

example: 'mfs', 'cont', 'cube'

example: mode='mfs' produce one image per source and spw

mode='cont' produce one image per source and aggregate over all specified spws

mode='cube' produce an LSRK frequency cube, channels are specified in frequency

hm_cell -- Cell size (x, y)

default: " Compute cell size based on the UV coverage of all the fields to be imaged and use a 5 pix per beam sampling. The pix per beam specification uses the above default cell size ('5ppb') and scales it accordingly.

example: ['0.5arcsec', '0.5arcsec'] '3ppb'

hm_imsz -- Image X and Y size in pixels or PB level for single field. The explicit sizes must be even and divisible by 2,3,5,7 only.

default: " The default values are derived as follows:

1. Determine phase center and spread of field centers around it.

2. Set the size of the image to cover the spread of field centers plus a border of width 0.75 * beam radius, to first null.

3. Divide X and Y extents by cell size to arrive at the number of pixels required. The PB level setting for single fields leads to an imsize extending to the specified level plus 5% padding in all directions.

example: [120, 120], '0.3pb'

calmaxpix -- Maximum image X or Y size in pixels if a calibrator is being imaged ('PHASE', 'BANDPASS', 'AMPLITUDE' or 'FLUX').

default: 300

example: 300

width -- Output channel width.

default: " Difference in frequency between first 2 selected channels. for frequency mode images.

example: '24.2kHz' 'pilotimage' for 15 MHz / 8 channel heuristic

nbins -- Channel binning factors per spw

default: " Binning factors for each spw. Format: 'spw1:nb1,spw2:nb2,...'

Optional wildcard: '*:nb'

example: '9:2,11:4,13:2,15:8' '*:2'

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the h_init or hif_importdata sets.

default: " use all measurement sets in the context

example: 'ngc5921.ms', ['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

intent -- Select intents for which associated fields will be imaged.

default: 'TARGET'

example: 'PHASE,BANDPASS'

field -- Select fields to image. Use field name(s) NOT id(s). Mosaics are assumed to have common source / field names. If intent is specified only fields with data matching the intent will be selected. The fields will be selected from measurement sets in 'vis'.

default: " Fields matching matching intent, one image per target source.

example: '3C279', 'Centaurus*', '3C279,J1427-421'

spw -- Select spectral window/channels to image.

default: " Individual images will be computed for all science spectral windows.

example: '9'

contfile -- Name of file with frequency ranges to use for continuum images.

default: 'cont.dat'

example: 'mycont.dat'

linesfile -- Name of file with line frequency ranges to exclude for continuum images.

default: 'lines.dat'

example: 'mylines.dat'

uvrange -- Select a set of uv ranges ro image.

default: " All uv data is included

example: '0~1000klambda', ['0~100klambda', 100~1000klambda]

phasecenter -- Direction measure or field id of the image center.

default: " The default phase center is set to the mean of the field directions of all fields that are to be image together.

example: 0, 'J2000 19h30m00 -40d00m00'

nchan -- Total number of channels in the output image(s)

default: -1 Selects enough channels to cover the data selected by spw consistent with start and width.

example: 100

start -- First channel for frequency mode images.

default " Starts at first input channel of the spw.

example: '22.3GHz'

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Examples

1. 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

Table 27: hif_makeimlist default settings

| Parameter | Type | Default | Description |
|--------------------|-------------|---------|--|
| vis | stringArray | None | List of input measurement sets |
| imagename | string | None | Prefix for output image names, '\\' for default. |
| intent | string | TARGET | Set of data selection intents |
| field | string | None | Set of data selection field names or ids, '\\' for all |
| spw | string | None | Set of data selection spectral window/channels, '\\' for all |
| contfile | string | None | Name of file with frequency ranges to use for continuum images. |
| linesfile | string | None | Name of file with line frequency ranges to exclude for continuum images. |
| uvrange | string | None | Set of data selection uv ranges, '\\' for all. |
| specmode | string | None | Spectral gridding type (mfs, cont, cube, '\\' for default) |
| outframe | string | None | Velocity frame of output image (LSRK, '\\' for default) |
| hm_imsz | stringArray | None | Image X and Y size(s) in pixels or PB level (single fields), '\\' for default. Single value same for both. '\\pb\\' for PB level. |
| hm_cell | stringArray | None | Image X and Y cell size(s) with units or pixels per beam, '\\' for default. Single value same for both. '\\ppb\\' for pixels per beam. |
| calmaxpix | int | 300 | Maximum X and Y size of calibrator images in pixels |
| phasecenter | any | None | Image center (direction or field index, '\\' for default) |
| nchan | int | -1 | Number of channels, -1 = all |
| start | any | None | Channel start, '\\' for default |
| width | any | None | Channel width, '\\' for default. |

| | | | |
|----------------------|--------|-----------|--|
| nbins | any | None | Channel binning factors per spw, '\\' for default. |
| 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 |

6.20 hif_mstransform

Create new measurement sets for imaging from the corrected column of the input measurement set. By default all science target data is copied to the new ms. The new measurement set is not re-indexed to the selected data in the new ms will have the same source, field, and spw names and ids as it does in the parent ms.

Task Description

Select data from calibrated MS(s) to form new MS(s) for imaging
Create a list of science target MS(s) for imaging

Keyword Arguments

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

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

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

vis -- The list of input measurement sets to be transformed. Defaults to the list of measurement sets specified in the pipeline import data task.
default "": Transform all calibration measurement sets in the context.
example: 'ngc5921.ms', ['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

outputvis -- The list of output transformed measurement sets to be used for imaging. The output list must be the same length as the input list.
default "", The output name defaults to `_target.ms`
example: 'ngc5921.ms', ['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

field -- Select fields name(s) or id(s) to transform. Only fields with data matching the intent will be selected.
default: "", Fields matching matching intent.
example: '3C279', 'Centaurus*', '3C279,J1427-421'

intent -- Select intents for which associated fields will be imaged.
default: "", Only TARGET data is selected.

example: 'PHASE,BANDPASS'

spw -- Select spectral window/channels to image.

default: ", All science spws for which the specified intent is valid are selected

example: '9'

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Examples

1. Create a science target ms from the corrected column in the input ms.

```
hif_mstransform()
```

2. Make a phase and bandpass calibrator targets ms from the corrected column in the input ms.

```
hif_mstransform(intent='PHASE,BANDPASS')
```

Parameter List

Table 28: hif_mstransform default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|-----------|---|
| vis | stringArray | None | The list of input measurement sets |
| outputvis | stringArray | None | The list of transformed measurement sets to be used for imaging |
| field | string | None | Set of data selection field names or ids, '*' for all |
| intent | string | None | Set of data selection intents |
| spw | string | None | Set of data selection spectral window ids '*' for all |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results to the pipeline context |

6.21 hif_normflux

Derive flux densities for point source transfer calibrators using flux models for reference calibrators. Flux values are determined by:

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

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

Task Description

Average calibrator fluxes across measurement sets

Derive flux densities for point source transfer calibrators using flux models for reference calibrators.

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

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

phaseupsolint -- Time solution intervals in CASA syntax for the phase solution.
default: 'inf'
example: 'inf', 'int', '100sec'

solint -- Time solution intervals in CASA syntax for the amplitude solution.
default: 'inf'
example: 'inf', 'int', '100sec'

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context
default: ""
example: ['M32A.ms', 'M32B.ms']

reference -- A string containing a comma delimited list of field names defining the reference calibrators. Defaults to field names with intent '*AMP*'.
default: ""
example: 'M82,3C273'

transfer -- A string containing a comma delimited list of field names defining the transfer calibrators. Defaults to field names with intent '*PHASE*'.

default: "

example: 'J1328+041,J1206+30'

refintent -- A string containing a comma delimited list of intents used to select the reference calibrators. Defaults to *AMP*.

default: "

example: ", '*AMP*'

refspwmap -- Vector of spectral window ids enabling scaling across spectral windows. Defaults to no scaling

default: [-1]

example: [1,1,3,3] (4 spws, reference fields in 1 and 3, transfer fields in 0,1,2,3)

transintent -- A string containing a comma delimited list of intents defining the transfer calibrators. Defaults to *PHASE*.

default: "

example: ", '*PHASE*'

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Issues

Should we add a spw window selection option here?

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

Examples

1. Compute flux flux values for the phase calibrator using model data from the amplitude calibrator.
hif_gfluxscale ()

Parameter List

Table 29: hif_normflux default settings

| Parameter | Type | Default | Description |
|----------------------|---------|-----------|---|
| vis | string | None | List of input measurements sets |
| refintent | string | None | Observing intent of reference fields |
| transintent | string | None | Observing intent of transfer fields |
| reference | variant | None | Reference calibrator field name(s) |
| transfer | variant | None | Transfer calibrator field name(s) |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display commands (True) |
| acceptresults | bool | True | Automatically accept results into context |

6.22 hif_rawflagchans

Task Description

Flag deviant baseline/channels in raw data

hif_rawflagchans flags deviant baseline/channels in the raw data.

The flagging views used are derived from the raw data for the specified intent - default is BANDPASS.

Bad baseline/channels are flagged for all intents, not just the one that is the basis of the flagging views.

For each spectral window the flagging view is a 2d image with axes 'channel' and 'baseline'. The pixel for each channel,baseline is the time average of the underlying unflagged raw data.

The baseline axis is labelled by numbers of form id1.id2 where id1 and id2 are the IDs of the baseline antennas. Both id1 and id2 run over all antenna IDs in the observation. This means that each baseline is shown twice but has the benefit that 'bad' antennas are easily identified by eye.

Three flagging methods are available:

If parameter flag_hilo is set True then outliers from the median of each flagging view will be flagged.

If parameter flag_bad_quadrant is set True then a simple 2 part test is used to check for bad antenna quadrants and/or bad baseline quadrants. Here a 'quadrant' is defined simply as one quarter of the channel axis. The first part of the test is to note as 'suspect' those points further from the view median than fbq_hilo_limit * MAD. The second part is to flag entire antenna/quadrants if their fraction of suspect points exceeds fbq_antenna_frac_limit. Failing that, entire baseline/quadrants may be flagged if their fraction of suspect points exceeds fbq_baseline_frac_limit. Suspect points are not flagged unless as part of a bad antenna or baseline quadrant.

Keyword Arguments

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

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

flag_hilo -- True to flag channel/baseline data further from the view median than $fhl_limit * MAD$.
default: True

fhl_limit -- If flag_hilo is True then flag channel/baseline data further from the view median than $fhl_limit * MAD$.
default: 20

fhl_minsample -- Do no flagging if the view median and MAD are derived from fewer than fhl_minsample view pixels.
default: 5

flag_bad_quadrant -- True to search for and flag bad antenna quadrants and baseline quadrants. Here a '/quadrant/' is one quarter of the channel axis.
default: True

fbq_hilo_limit -- If flag_bad_quadrant is True then channel/baselines further from the the view median than $fbq_hilo_limit * MAD$ will be noted as 'suspect'. If there are enough of them to indicate that an antenna or baseline quadrant is bad then all channel/baselines in that quadrant will be flagged.
default: 8.0

fbq_antenna_frac_limit -- If flag_bad_quarant is True and the fraction of suspect channel/baselines in a particular antenna/quadrant exceeds fbq_antenna_frac_limit then all data for that antenna/quadrant will be flagged.
default: 0.2

fbq_baseline_frac_limit -- If flag_bad_quarant is True and the fraction of suspect channel/baselines in a particular baseline/quadrant exceeds fbq_baseline_frac_limit then all data for that baseline/quadrant will be flagged.
default: 1.0 (i.e. no flagging)

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

vis -- List of input measurement sets.
default: [] - Use the measurement sets currently known to the pipeline context.

intent -- A string containing the list of intents to be checked for antennas with deviant gains. The default is blank, which causes the task to select the 'BANDPASS' intent.
default: ""
example: '*BANDPASS*'

spw -- The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context.

default: ""

example: '17', '11, 15'

-- pipeline task execution modes

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

default: True

acceptresults -- This parameter has no effect. The Tsyscal file is already in the pipeline context and is flagged in situ.

Output

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

Examples

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

hif_rawflagchans()

equivalent to: hif_rawflagchans(flag_hilo=True, fhl_limit=20, flag_bad_quadrant=True,

fbq_hilo_limit=8, fbq_antenna_frac_limit=0.2, fbq_baseline_frac_limit=1.0)

Parameter List

Table 30: hif_rawflagchans default settings

| Parameter | Type | Default | Description |
|-------------------------|-------------|-----------|--|
| vis | stringArray | None | List of input measurement sets |
| spw | string | None | Set of data selection spectral windows, '\\' for all |
| intent | string | None | Data intent to use in creating flagging view |
| flag_hilo | bool | True | True to flag outlier baseline/channels |
| fhl_limit | double | 20.0 | Flag baseline/channels further from median than limit * MAD |
| fhl_minsample | double | 5 | Minimum number of points in sample |
| flag_bad_quadrant | bool | True | True to flag bad quadrants in antennas |
| fbq_hilo_limit | double | 8.0 | Note as 'suspect' baseline/channels further from median than limit * MAD |
| fbq_antenna_frac_limit | double | 0.2 | Flag antenna quadrant if it's fraction of 'suspect' baseline/channels > limit |
| fbq_baseline_frac_limit | double | 1.0 | Flag baseline quadrant if it's fraction of 'suspect' baseline/channels > limit |
| pipelinemode | string | automatic | The pipeline operations mode |

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

6.23 hif_refant

The hif_refant task selects a list of reference antennas and stores them in the pipeline context in priority order. The priority order is determined by a weighted combination of scores derived by the antenna selection heuristics. In manual mode the reference antennas can be set by hand.

Task Description

Select the best reference antennas

The hif_refant task selects a list of reference antennas and outputs them in priority order. The priority order is determined by a weighted combination of scores derived by the antenna selection heuristics.

Keyword Arguments

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

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

hm_refant -- The heuristics method or mode for selection the reference antenna. The options are 'manual' and 'automatic. In manual mode a user supplied reference antenna refant is supplied. In 'automatic' mode the antennas are selected automatically.
default: 'automatic'

refant -- The user supplied reference antenna for 'manual' mode. If no antenna list is supplied an empty list is returned.
default: ""
example: 'DV05'

geometry -- Score antenna by proximity to the center of the array. This option is quick as only the ANTENNA table must be read.
default: True

flagging -- Score antennas by percentage of unflagged data. This option requires computing flagging statistics.
default: True

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the pipeline context.
default: ""

example: ['M31.ms']

field -- The list of field names or field ids for which flagging scores are computed if hm_refant='automatic' and flagging = True
default: "
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to all supported intents.
default: "
example: '*BANDPASS*'

spw -- The list of spectral windows and channels for which flagging scores are computed if hm_refant='automatic' and flagging = True.
default: "
example: '11,13,15,17'

--- pipeline task execution modes

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

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

Output

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

Examples

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

Parameter List

Table 31: hif_refant default settings

| Parameter | Type | Default | Description |
|-----------|-------------|-----------|---|
| vis | stringArray | None | List of input measurement sets |
| field | string | None | Set of data selection field names or ids |
| spw | string | None | Set of data selection spectral windows/channels |
| intent | string | None | Set of data selection intents |
| hm_refant | string | automatic | The referance antenna heuristics mode |
| refant | string | None | List of reference antennas |

| | | | |
|----------------------|--------|-----------|--|
| geometry | bool | True | Score by proximity to center of the array |
| flagging | bool | True | Score by percentage of good data |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

6.24 hif_restoredata

The hif_restoredata restores flagged and calibrated data from archived ASDMs and pipeline flagging and calibration data products. Pending archive retrieval support hif_restore data assumes that the required products are available in the rawdata_dir in the format produced by the hif_exportdata task.

hif_restoredata assumes that the following entities are available in the raw data directory:

- the ASDMs to be restored
- for each ASDM in the input list
- a compressed tar file of the final flagversions file, e.g. uid__A002_X30a93d_X43e.ms.flagversions.tar.gz
- a text file containing the applycal instructions, e.g. uid__A002_X30a93d_X43e.ms.calapply.txt
- a compressed tar file containing the caltables for the parent session, e.g. uid__A001_X74_X29.session_3.caltables.tar.gz

hif_restore data performs the following operations:

- imports the ASDM(s)
- removes the default MS.flagversions directory created by the filler
- restores the final MS.flagversions directory stored by the pipeline
- restores the final set of pipeline flags to the MS
- restores the final calibration state of the MS
- restores the final calibration tables for each MS
- applies the calibration tables to each MS

Task Description

Restore flagged and calibration interferometry data from a pipeline run

The hif_restoredata task restores flagged and calibrated measurements sets from archived ASDMs and pipeline flagging and calibration data products.

Keyword Arguments

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

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

default: 'automatic'.

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

vis -- List of raw visibility data files to be restored. Assumed to be in the directory specified by rawdata_dir.

default: None

example: vis=['uid___A002_X30a93d_X43e']

session -- List of sessions one per visibility file.

default: []

example: session=['session_3']

products_dir -- Name of the data products directory. Currently not used.

default: './products'

example: products_dir='myproductspath'

rawdata_dir -- Name of the rawdata subdirectory.

default: './rawdata'

example: rawdata_dir='myrawdatapath'

lazy -- Use the lazy filler option

default: False

example: lazy=True

bdfflags -- Set the BDF flags

default: True

example: bdfflags=False

ocorr_mode -- Set ocorr_mode

default: 'ca'

example: ocorr_mode='ca'

--- 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. Restore the pipeline results for a single ASDM in a single session

hif_restoredata (vis=['uid___A002_X30a93d_X43e'], session=['session_1'], ocorr_mode='ca')

Parameter List

Table 32: hif_restoredata default settings

| Parameter | Type | Default | Description |
|---------------|-------------|-------------|--|
| vis | stringArray | None | List of input visibility data |
| session | stringArray | None | List of sessions one per visibility file |
| products_dir | string | ../products | The archived pipeline data products directory |
| copytoraw | bool | True | Copy calibration and flagging tables to raw data directory |
| rawdata_dir | string | ../rawdata | The rawdata directory |
| lazy | bool | False | Use the lazy filler option |
| bdfflags | bool | True | Set the BDF flags |
| ocorr_mode | string | ca | ALMA default set to ca |
| 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 |

6.25 hif_setjy

Task Description

Fill the model column with calibrated visibilities
Fills the model column with the model visibilities.

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

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

fluxdensity -- Specified flux density [I,Q,U,V] in Jy. Uses [1,0,0,0] flux density for unrecognized sources, and standard flux densities for ones recognized by 'standard', including 3C286, 3C48, 3C147, and several planets, moons, and asteroids.
default=-1
example: [3.06,0.0,0.0,0.0]

reffile -- Path to a file containing flux densities for calibrators unknown to CASA. Values given in this file take precedence over the CASA-derived values for all calibrators except solar system

calibrators. By default the path is set to the CSV file created by `h_importdata`, consisting of catalogue fluxes extracted from the ASDM.

default: ""

example: "", 'working/flux.csv'

spix -- Spectral index for fluxdensity $S = \text{fluxdensity} * (\text{freq}/\text{reffreq})^{**\text{spix}}$ Only used if fluxdensity is being used. If fluxdensity is positive, and spix is nonzero, then reffreq must be set too. It is applied in the same way to all polarizations, and does not account for Faraday rotation or depolarization.

Default: 0

reffreq -- The reference frequency for spix, given with units. Provided to avoid division by zero. If the flux density is being scaled by spectral index, then reffreq must be set to whatever reference frequency is correct for the given fluxdensity and spix. It cannot be determined from vis. On the other hand, if spix is 0, then any positive frequency can be used and will be ignored.

Default: '1GHz'

Examples: '86.0GHz', '4.65e9Hz'

scalebychan -- This determines whether the fluxdensity set in the model is calculated on a per channel basis. If False then only one fluxdensity value is calculated per spw.

default: True

standard -- Flux density standard, used if fluxdensity[0] less than 0.0. The options are: 'Baars', 'Perley 90', 'Perley-Taylor 95', 'Perley-Taylor 99', 'Perley-Butler 2010' and 'Butler-JPL-Horizons 2010'.

default: 'Butler-JPL-Horizons 2012' for solar system object 'Perley-Butler 2010' otherwise

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets defined in the pipeline context.

default: []

field -- The list of field names or field ids for which the models are to be set. Defaults to all fields with intent '*AMPLITUDE*'

default: ""

example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to all data with amplitude intent.

default: ""

example: '*AMPLITUDE*'

spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.

default: ""

example: '11,13,15,17'

model -- Model image for setting model visibilities. Not fully supported.

default: ""

example: see details in help for CASA setjy task

--- pipeline task execution modes

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

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

Output

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

Issues

Support for the setjy spix parameter needs to be added.

Examples

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

Parameter List

Table 33: hif_setjy default settings

| Parameter | Type | Default | Description |
|--------------|-------------|-----------|--|
| vis | stringArray | None | List of input measurement sets |
| field | string | None | List of field names or ids |
| intent | string | None | Observing intent of flux calibrators |
| spw | string | None | List of spectral window ids |
| model | string | None | File location for field model |
| reffile | string | None | Path to file with fluxes for non-solar system calibrators |
| normfluxes | bool | False | Normalize lookup fluxes |
| reffreq | string | 1GHz | Reference frequency for spix |
| fluxdensity | any | -1 | Specified flux density [I,Q,U,V]; -1 will lookup values |
| spix | double | 0.0 | Spectral index of fluxdensity |
| scalebychan | bool | True | Scale the flux density on a per channel basis or else on a per spw basis |
| standard | variant | None | Flux density standard |
| pipelinemode | string | automatic | The pipeline operating mode |

| | | | |
|----------------------|------|-------|---|
| dryrun | bool | False | Run the task (False) or display the commands (True) |
| acceptresults | bool | True | Automatically accept results into the context |

6.26 hif_setmodels

Set model fluxes values for calibrator reference and transfer sources using lookup values. By default the reference sources are the flux calibrators and the transfer sources are the bandpass, phase, and check source calibrators. Reference sources which are also in the transfer source list are removed from the transfer source list.

Builtin lookup tables are used to compute models for solar system object calibrators. Point source models for other calibrators are provided in the reference file. Normalize fluxes are computed for transfer sources if the normfluxes parameter is set to True.

The reference file default to a file called 'flux.csv' in the current working directory. This file is normally created in the importdata step. The file is in 'csv' format and contains the following comma delimited columns.

vis,fieldid,spwid,I,Q,U,V,pix,comment

Task Description

Set calibrator source models

Derive flux densities for point source transfer calibrators using flux models for reference calibrators.

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

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

reffile -- The reference file containing a lookup table of point source models This file currently defaults to 'flux.csv' in the working directory. This file must conform to the standard pipeline 'flux.csv' format.
default: "
example: 'myfluxes.csv'

normfluxes -- Normalize the transfer field lookup fluxes to 1.0. The reference field fluxes are never normalized.
default: True
example: False

scalebychan -- Scale the flux density on a per channel basis or else on a per spw basis
default: True
example: False

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context
 default: "
 example: ['M32A.ms', 'M32B.ms']

reference -- A string containing a comma delimited list of field names defining the reference calibrators. Defaults to field names with intent 'AMPLITUDE'.
 default: "
 example: 'M82,3C273'

refintent -- A string containing a comma delimited list of intents used to select the reference calibrators. Defaults to 'AMPLITUDE'.
 default: 'AMPLITUDE'. " Means no sources.
 example: 'BANDPASS'

transfer -- A string containing a comma delimited list of field names defining the transfer calibrators. Defaults to field names with intent ".
 default: 'BANDPASS,PHASE,CHECK'
 example: 'J1328+041,J1206+30'

transintent -- A string containing a comma delimited list of intents defining the transfer calibrators. Defaults to 'BANDPASS,PHASE,CHECK'. " stands for no transfer sources.
 default: 'BANDPASS,PHASE,CHECK'
 example: 'PHASE'

--- pipeline task execution modes

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

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

Output

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

Examples

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

Parameter List

Table 34: hif_setmodels default settings

| Parameter | Type | Default | Description |
|-----------|--------|---------|---------------------------------|
| vis | string | None | List of input measurements sets |

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

6.27 hif_show_calstate

hif_show_calstate displays the current on-the-fly calibration state of the pipeline as a set of equivalent applycal calls.

Task Description

Show the current pipeline calibration state

Keyword arguments:

None

Parameter List

No parameter

6.28 hif_tclean

Task Description

Compute clean map
Compute a cleaned image for a particular target source/intent and spectral window.

Keyword Arguments

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

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

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the context.
default: "
example: vis=['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']

imagename -- Prefix of output images. Defaults to one of the following options depending on the availability of project information:
'{ousstatus uid}.{field}.[{intent}.]s{stage number}.spw{spw}'
'multivis.{field}.[{intent}.]s{stage number}.spw{spw}'

cleanboxes and thresholds to use as it goes. For each iteration the output images are:
{prename}.iter{n}.image; cleaned and restored image
{prename}.iter{n}.psf; point spread function (dirty beam)
{prename}.iter{n}.flux; relative sky sensitivity over field
{prename}.iter{n}.flux.pbcoverage; relative pb coverage over field (only for mosaics)
{prename}.iter{n}.model; image of clean components
{prename}.iter{n}.residual; image of residuals
{prename}.iter{n}.cleanmask; image of cleanmask used
default: "
example: 'test1'

intent -- An intent against which the selected fields are matched. Default means select all data from fields specified by 'field' parameter
default: "
example: ", 'TARGET'

field -- Fields id(s) or name(s) to image or mosaic. Must be set.
default:
example: '3C279', 'Centaurus'

spw -- Spectral window/channels to image. '\' for all science data.
default: "
example: '9', '9,11'

spwsel_lsrk -- Spectral window LSRK frequency selection for continuum. {} for all science data.
default: {}
example: {'spw16': '89.1~89.5GHz;90.2~90.3GHz', 'spw18': '101.2~102.1GHz'}

spwsel_topo -- Per MS spectral window TOPO frequency selection for continuum. [] for all science data.
default: {}
example: {'spw16:89.1~89.5GHz;90.2~90.3GHz'}

specmode -- Frequency imaging mode, 'mfs', 'cont', 'cube'. '\' defaults to 'cube' if intent parameter includes 'TARGET' otherwise 'mfs'.

default: "
example: 'mfs', 'cont', 'cube'

gridder -- Gridding options, 'standard', 'mosaic'. Derived as follows:

1. The 'field' parameter is converted into a list of field_ids for each measurement set in 'vis'.
2. If there is more than 1 field_id in the list for any measurement set then gridder is set to 'mosaic', otherwise it will be set to 'standard'.

default: " 'standard'

deconvolver -- Minor cycle algorithm e.g. hogbom or clark clean. \\'

defaults: 'hogbom'

outframe -- The reference frame of the output image. The only supported option is 'LSRK'

default: "

example: 'LSRK'

imsize -- X and Y image size in pixels). Must be even and contain factors 2,3,5,7 only.

Default derived as follows:

1. Determine 'phasecenter' value and spread of field centres around it.
2. Set size of image to cover spread of field centres plus a border of width $0.75 * \text{beam radius}$ (to first null).
3. Divide x and y extents by 'cell' values to arrive at the numbers of pixels required.

default: "

example: [320,320]

cell -- X and Y cell size. Derived from maximum UV spacing. Details TBD

default "

example: ['0.5arcsec', '0.5arcsec']

phasecenter -- Direction measure or field id for the mosaic center.

Default derived as follows:

1. Make an array containing all the field centers to be imaged together.
2. Derive the mean direction from the directions array.

default: \\'

example: 2

nchan -- Number of channels or planes in the output image, -1 for all

default: -1

example: 128

width -- Width of spectral dimension in frequency, \\' for default.

default: \\'

example: '7.8125MHz'

nbin -- Channel binning factor

default: -1

example: 2

weighting -- Weighting to apply to visibilities. Options are: 'natural', 'uniform', 'briggs', 'superuniform', 'briggsabs', 'radial'

default: 'briggs'

example: weighting='uniform'

robust -- Parameter for 'briggs' and 'briggsabs' weighting. Ranges from -2.0 to 2.0. -2 for uniform +2 for natural.
default=-999.0
example: 0.7

noise -- Parameter for 'briggsabs' weighting
default: '1.0Jy'
example: '0.5Jy'

npixels -- Parameter for 'briggs' and 'briggsabs'; weighting
default: 0
example: 0

restoringbeam -- Gaussian restoring beam for clean, '\'' for default
default: '\'
example:

hm_masking -- Clean masking mode. Options are 'centralregion', 'psf', 'psfiter', 'auto', 'auto-thresh', 'manual' and 'none'
default: 'centralregion'
example: 'manual'

hm_maskthreshold -- Auto-boxing mask threshold. See tclean for options.

mask -- Image mask for hm_masking manual mode. User responsible for matching image sizes, coordinates, etc.
default: '\'
example: 'mymask.mask'

niter -- Maximum number of iterations per clean call
default: 5000
example: 500

threshold -- Threshold for cleaning
default: '0.0'
example: '0.05'

maxncleans -- Maximum number of clean calls
default: 1
example: 10

cleanconranges -- Clean continuum frequency ranges in cubes
default=False
Options: False, True

subcontms -- Subtract continuum in MS
Caveat: if enabled, repeated continuum subtraction will lead to erroneous negative fluxes.
default=False
Options: False, True

parallel -- use MPI cluster to clean images

default: 'automatic'

--- pipeline task execution modes

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

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

Output

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

Examples

1. Make an 'mfs' image of calibrator 3c279 using data in spectral window 1. The cell size is set to 0.2 arcsec in RA and Dec. Other clean parameters are derived from heuristics:
`hif_tclean(field='3c279', cell='0.2arcsec', spw='1', specmode='mfs')`

2. Make a cube of calibrator 3c279 using data in spectral window 1. The cube planes will be evenly spaced in frequency in the LSRK frame. Other clean parameters are derived from heuristics.
`hif_tclean(field='3c279', cell='0.2arcsec', spw='1', specmode='cube', outframe='LSRK')`

Parameter List

Table 35: hif_tclean default settings

| Parameter | Type | Default | Description |
|--------------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets, '\' |
| imagename | string | None | Prefix for image filenames, '\' |
| intent | string | None | Set of data selection intents, '\' |
| field | string | None | Set of data selection field names or ids |
| spw | string | None | Set of data selection spectral window/channels \' |
| spwsel_lsrk | any | {} | Dictionary of LSRK spw frequency selections, {} for automatic |
| spwsel_topo | any | {} | List of TOPO spw frequency selections per MS, [] for automatic |
| uvrange | any | None | Set of uv ranges, '\' |
| specmode | string | None | Spectral gridding type (mfs, cont, cube, '\' |
| gridder | string | None | Gridding options (standard, mosaic, '\' |

| | | | |
|-------------------------|-------------|---------------|--|
| deconvolver | string | None | Minor cycle algorithm (hogbom, clark, mtmfs, '\') for default) |
| outframe | string | None | Velocity frame of output image (LSRK, '\') for default) |
| imsize | intArray | None | X and Y image size in pixels, single value same for both, '\') for default |
| cell | stringArray | None | X and Y cell size(s), single value same for both, '\') for default |
| phasecenter | any | None | Image center (direction or field index), '\') for default |
| nchan | int | -1 | Number of channels or planes in output image, -1 = all |
| start | any | None | Start of output spectral dimension |
| width | any | None | Width of output spectral channels, '\') for default |
| nbin | int | -1 | Channel binning factor, -1 for default. |
| weighting | string | briggs | Type of weighting |
| robust | double | -999.0 | Briggs weighting robustness parameter |
| noise | any | 1.0Jy | Briggs weighting noise parameter |
| npixels | int | 0 | Weighting algorithm parameter |
| restoringbeam | stringArray | None | Gaussian restoring beam, '\') for default |
| hm_masking | string | centralregion | Pipeline heuristics masking option |
| hm_maskthreshold | string | None | Auto-boxing mask threshold |
| hm_cleaning | string | rms | Pipeline clean control heuristics |
| mask | any | None | User mask, '\') for whole image |
| niter | int | 5000 | Maximum number of clean iterations |
| threshold | double | 0.0 | Flux level to stop cleaning, must include units: '\1.0mJy' |
| tlimit | double | 4.0 | Times the sensitivity limit for cleaning |
| masklimit | int | 4 | Times good mask pixels for cleaning |
| maxncleans | int | 1 | Maximum number of clean task calls |
| cleanconranges | bool | False | Clean continuum frequency ranges in cubes |
| subcontms | bool | False | Subtract continuum in MS |
| parallel | string | automatic | Clean images using MPI cluster |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display the command (True) |

| | | | |
|----------------------|------|------|---|
| acceptresults | bool | True | Add the results to the pipeline context |
|----------------------|------|------|---|

6.29 hif_uvcontfit

Task Description

Fit the continuum in the UV plane

This task estimates the continuum emission by fitting polynomials to the real and imaginary parts of the spectral windows and channels selected by spw and exclude spw. This fit represents a model of the continuum in all channels. Fit orders less than 2 are strongly recommended. Spw window combination is not currently supported.

Keyword Arguments

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

default: 'automatic'.

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

solint --

default: 'int'

example: solint='30s'

fitorder -- The fit order of the polynomials.

default: 1

example: fitorder = 0

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

vis -- The list of input measurement sets for which the UV continuum fits are to be generated.

Defaults to the list of imaging measurement sets specified in the pipeline import data task.

default "": Compute the continuum fit for all calibration measurement sets in the context.

example: 'ngc5921.ms', ['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

caltable -- The list of output calibration tables one per input MS.

default: "", The output name defaults to the standard pipeline name scheme

example: ['M51.uvcont']

contfile -- The file containing the continuum frequency ranges used for the continuum fit.

default: "", Defaults first to the file named in the context, next to a file called 'cont.dat' in the pipeline working directory, or ""

example: contfile = 'mycontfile'

field -- The list of field names or field ids for which UV continuum fits are computed. Defaults to all fields.

default: ""

example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched.
 default: "", Defaults to all data with TARGET intent.
 example: 'PHASE'

spw -- The list of spectral windows and channels for which uv continuum fits are computed.
 default: "", Defaults to all science spectral windows.
 example: '11,13,15,17'

combine -- Data axes to be combined for solving. Axes are 'scan', 'spw', or "". This option is currently not supported.
 default: "", None.
 example: combine= 'scan'

Parameter List

Table 36: hif_uvcontfit default settings

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

6.30 hif_uvcontsub

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

Task Description

Subtract the fitted continuum from the data
Apply the UV continuum fit stored in the callibrary to the data.

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

applymode -- Calibration apply mode `"='calflagstrict'`: calibrate data and apply flags from solutions using the strict flagging convention `'trial'`: report on flags from solutions, dataset entirely unchanged `'flagonly'`: apply flags from solutions only, data not calibrated `'calonly'`: calibrate data only, flags from solutions NOT applied `'calflagstrict'`:
`'flagonlystrict'`: same as above except flag spws for which calibration is unavailable in one or more tables (instead of allowing them to pass uncalibrated and unflagged)
default: "", defaults to 'calonly'

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

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the pipeline context.
default: []
example: ['X227.ms']

field -- A string containing the list of field names or field ids to which the calibration will be applied. Defaults to all fields in the pipeline context.
default: ""
example: '3C279', '3C279, M82'

intent -- A string containing a the list of intents against which the selected fields will be matched. Defaults to all supported intents in the pipeline context.
default: ""
example: '*TARGET*'

spw -- The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context.
default: ""
example: '17', '11, 15'

--- pipeline task execution modes

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

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

Output

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

Examples

1. Apply the calibration to the target data
hif_uvcontsub()

Parameter List

Table 37: hif_uvcontsub default settings

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

7 Interferometry ALMA Task Descriptions

7.1 hifa_antpos

The hifa_antpos task corrects the antenna positions recorded in the ASDMs using updated antenna position calibration information determined after the observation was taken. Corrections can be input by hand, read from a file on disk, or in future by querying an ALMA database service. The antenna positions file is in 'csv' format containing 6 comma delimited columns as shown below. The default name of this file is 'antennapos.csv'

List if sample antennapos.csv file

```
ms,antenna,xoffset,yoffset,zoffset,comment  
uid___A002_X30a93d_X43e.ms,DV11,0.000,0.010,0.000,"No comment"
```

uid___A002_X30a93d_X43e.dup.ms,DV11,0.000,-0.010,0.000,"No comment"

The corrections are used to generate a calibration table which is recorded in the pipeline context and applied to the raw visibility data, on the fly to generate other calibration tables, or permanently to generate calibrated visibilities for imaging.

Task Description

Derive an antenna position calibration table

Keyword Arguments

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

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

hm_antpos -- Heuristics method for retrieving the antenna position corrections. The options are 'online' (not yet implemented), 'manual', and 'file'.
default: 'file'
example: hm_antpos='manual'

antenna -- The list of antennas for which the positions are to be corrected if hm_antpos is 'manual'
default: none
example 'DV05,DV07'

offsets -- The list of antenna offsets for each antenna in 'antennas'. Each offset is a set of 3 floating point numbers separated by commas, specified in the ITRF frame.
default: none
example: [0.01, 0.02, 0.03, 0.03, 0.02, 0.01]

antposfile -- The file(s) containing the antenna offsets. Used if hm_antpos is 'file'. The default file name is 'antennapos.csv'

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

vis -- List of input visibility files
default: []
example: ['ngc5921.ms']

caltable -- Name of output gain calibration tables
default: []
example: caltable=['ngc5921.gcal']

-- Pipeline task execution modes

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

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

Output

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

Issues

The hm_antpos 'online' option will be implemented when the observing system provides an antenna position determination service.

Example

1. Correct the position of antenna 5 for all the visibility files in a single pipeline run.

hifa_antpos (antenna='DV05', offsets=[0.01, 0.02, 0.03])

2. Correct the position of antennas for all the visibility files in a single pipeline run using antenna positions files on disk. These files are assumed to conform to a default naming scheme if 'antposfile' is unspecified by the user.

hifa_antpos (hm_antpos='myantposfile.csv')

Parameter List

Table 38: hifa_antpos default settings

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

7.2 hifa_bandpass

hif_bandpass computes a bandpass solution for every specified science spectral window. By default a 'phaseup' pre-calibration is performed and applied on the fly to the data, before the

bandpass is computed. The hif_refant task may be used to precompute a prioritized list of reference antennas.

Task Description

Compute bandpass calibration solutions
Compute amplitude and phase as a function of frequency for each spectral window in each measurement set.
Previous calibration can be applied on the fly.

Keyword Arguments

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

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

hm_phaseup -- The pre-bandpass solution phaseup gain heuristics. The options are 'snr' (compute solution required to achieve the specified SNR), 'manual' (use manual solution parameters), and '' (none).
default: 'snr'
example: hm_phaseup='manual'

phaseupsolint -- The phase correction solution interval in CASA syntax. Used when hm_phaseup='manual' or as a default if the hm_phaseup='snr' heuristic computation fails.
default: 'int'
example: phaseupsolint='300s'

phaseupbw -- Bandwidth to be used for phaseup. Defaults to 500MHz. Used when hm_phaseup='manual'.
default: ''
example: '' default to entire bandpass, '500MHz' use centreal 500MHz

phaseupsnr -- The required SNR for the phaseup solution. Used only if hm_phaseup='snr'
default: 20.0
example: phaseupsnr=10.0

phaseupnsols -- The minimum number of phaseup gain solutions. Used only if hm_phaseup='snr'.
default: 2
example: phaseupnsols=4

hm_bandpass -- The bandpass solution heuristics. The options are 'snr' (compute the solution required to achieve the specified SNR), 'smoothed' (simple smoothing heuristics), and 'fixed' (use the user defined parameters for all spws).

solint -- Time and channel solution intervals in CASA syntax.
default: 'inf' Used for hm_bandpass='fixed', and as a default for the 'snr' and 'smoothed' options.
default: 'inf,7.8125MHz'
example: solint='inf,10ch', solint='inf'

maxchannels -- The bandpass solution smoothing factor in channels. The solution interval is bandwidth / 240. Set to 0 for no smoothing. Used if hm_bandpass='smoothed'.

default: 240

example: 0

evenbpints -- Force the per spe frequency solint to be evenly divisible into the spw bandpass if hm_bandpass='snr'

default: True

example: evenbpints=False

bpsnr -- The required SNR for the bandpass solution. Used only if hm_bandpass='snr'

default: 50.0

example: bpsnr=20.0

bpnsols -- The minimum number of bandpass solutions. Used only if hm_bandpass='snr'.

default: 8

hm_bandtype -- The type of bandpass. The options are 'channel' and 'polynomial' for CASA bandpass types = 'B' and 'BPOLY' respectively.

combine -- Data axes to combine for solving. Axes are ", 'scan','spw','field' or any comma-separated combination.

default: 'scan'

example: combine='scan,field'

minblperant -- Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions. Used for hm_bandtype='channel' only.

default: 4

minsnr -- Solutions below this SNR are rejected. Used for hm_bandtype='channel' only

default: 3.0

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context.

default: ""

example: ['M51.ms']

caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.

default: ""

example: ['M51.bcal']

field -- The list of field names or field ids for which bandpasses are computed. Defaults to all fields.

default: ""

example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to all data with bandpass intent.

default: ""

example: '*PHASE*'

spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.

default: "

example: '11,13,15,17'

refant -- Reference antenna names. Defaults to the value(s) stored in the pipeline context. If undefined in the pipeline context defaults to the CASA reference antenna naming scheme.

default: "

example: refant='DV01', refant='DV06,DV07'

solnorm -- Normalise the bandpass solutions

default: False

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Issues

There is currently some discussion about whether or not to do an 'ampup' operations at the same time as the 'phaseup'. This is not required for the bandpass computation but the amplitude information may provide a useful quality assessment measure.

The specified minsnr parameter is currently applied to the bandpass solution computation but not the 'phaseup' computation. Some noisy solutions in the phaseup may not be properly rejected.

Examples

1. Compute a channel bandpass for all visibility files in the pipeline context using the CASA reference antenna determination scheme.

```
hif_bandpass()
```

2. Same as the above but precompute a prioritized reference antenna list

```
hif_refant()
```

```
hif_bandpass()
```

Parameter List

Table 39: hifa_bandpass default settings

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

7.3 hifa_bpsolint

The phaseup gain time and bandpass frequency intervals are determined as follows.

- For each data set the list of source(s) to use for bandpass solution signal to noise estimation is compiled based on the values of the field, intent, and spw parameters.
- Source fluxes are determined for each spw and source combination.
- Fluxes in Jy are derived from the pipeline context.
- Pipeline context fluxes are derived from the online flux calibrator catalog, the ASDM, or the user via the flux.csv file.
- If no fluxes are available the task terminates.
- Atmospheric calibration and observations scans are determined for each spw and source combination.
- If intent is set to 'PHASE' are there are no atmospheric scans associated with the 'PHASE' calibrator, 'TARGET' atmospheric scans will be used in stead.
- If atmospheric scans cannot be associated with any of the spw and source combinations the task terminates.
- Science spws are mapped to atmospheric spws for each science spw and source combinations.
- If mappings cannot be determined for any of the spws the task terminates
- The median Tsys value for each atmospheric spw and source combination is determined from the SYSCAL table. Medians are computed first by channel, then by antenna, in order to reduce sensitivity to deviant values.
- The science spw parameters, exposure time(s), and integration time(s) are determined.
- The phase up time interval, in time units and number of integrations required to meet the phaseupsnr are computed, along with the phaseup sensitivity in mJy and the signal to noise per integration. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
- Warnings are issued if estimated phaseup gain time solution would contain fewer than minphasupints solutions
- The frequency interval, in MHz and number of channels required to meet the bpsnr are computed, along with the per channel sensitivity in mJy and the per channle signal to noise. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
- Warnings are issued if estimated bandpass olution would contain fewer than minbpnchan solutions

Task Description

Compute optimal bandpass calibration solution intervals

Compute the bandpass phaseup time solution interval and bandpass frequency solution interval.

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

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

phaseupsnr -- The required phaseup gain time interval solution signal to noise
default: 20.0

example: phaseupsnr = 10.0

minphaseupints -- The minimum number of time intervals in the phaseup gain solution.

default: 2

example: minphaseupints=4

bpnsr -- The required bandpass frequency interval solution signal to noise

default: 50.0

example: phaseupsnr = 20.0

minbpnchan -- The minimum number of frequency intervals in the bandpass solution.

default: 8

example: minbpnchans=16

hm_nantennas -- The heuristics for determines the number of antennas to use in the signal to noise estimate. The options are 'all' and 'unflagged'. The 'unflagged' options is not currently supported.

default: 'all'

example: hm_nantennas='unflagged'

maxfracflagged -- The maximum fraction of an antenna that can be flagged before its is excluded from the signal to noise estimate.

default: 0.90

example: maxfracflagged=0.80

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context.

default: "

example: ['M82A.ms', 'M82B.ms']

field -- The list of field names of sources to be used for signal to noise estimation. Defaults to all fields with the standard intent.

default: "

example: '3C279'

intent -- A string the intent against which the the selected fields are matched.

default: 'BANDPASS'

example: intent='PHASE'

spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all the science spectral windows for which there are both 'intent' and TARGET intents.

default: "

example: '13,15'

--- pipeline task execution modes

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

default: False

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

Output

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

Examples

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

hifa_bpsolint()

Parameter List

Table 40: hifa_bpsolint default settings

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

7.4 hifa_flagdata

The hifa_flagdata data performs basic flagging operations on a list of measurements including:

- applying online flags
- apply a flagging template
- autocorrelation data flagging
- shadowed antenna data flagging
- scan based flagging by intent or scan number
- edge channel flagging

Task Description

Do basic flagging

The `hifa_flagdata` data performs basic flagging operations on a list of measurement sets.

Keyword Arguments

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

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

autocorr -- Flag autocorrelation data.
default: True

shadow -- Flag shadowed antennas.
default: True

scan -- Flag a list of specified scans.
default: True

scannumber -- A string containing a comma delimited list of scans to be flagged.
example: '3,5,6'
default: ''

intents -- A string containing a comma delimited list of intents against which the scans to be flagged are matched.
example: '*BANDPASS*'
default: 'POINTING, FOCUS, ATMOSPHERE, SIDEBAND'

edgespw -- Flag the edge spectral window channels.
default: True

fracspw -- Fraction of the baseline correlator TDM edge channels to be flagged.
default: 0.0625

fracspwfps -- Fraction of the ACS correlator TDM edge channels to be flagged.
default: 0.48387

online -- Apply the online flags.
default: True

fileonline -- File containing the online flags. These are computed by the h_init or hif_importdata data tasks. If the online flags files are undefined a name of the form 'msname_flagonline.txt' is assumed.
default: "

template -- Apply flagging templates
default: True

filetemplate -- The name of an text file that contains the flagging template for RFI, birdies, telluric lines, etc. If the template flags files is undefined a name of the form 'msname_flagtemplate.txt' is assumed.
default: "

hm_tbuff -- The heuristic for computing the default time interval padding parameter. The options are 'halfint' and 'manual'. In 'halfint' mode tbuff is set to half the maximum of the median integration time of the science and calibrator target observations. The value of 0.048 seconds is subtracted from the lower time limit to accomodate the behavior of the ALMA Control system.
default: 'halfint'

tbuff -- The time in seconds used to pad flagging command time intervals if hm_tbuff='manual'. The default in manual mode is no flagging
default: [0.0,0.0]

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

vis -- The list of input measurement sets. Defaults to the list of measurement defined in the pipeline context.
default: "

flagbackup -- Back up any pre-existing flags.
default: False

--- pipeline task execution modes

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

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

Output

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

Examples

1. Do basic flagging on a measurement set
hifa_flagdata()

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

Parameter List

Table 41: hifa_flagdata default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|---|--|
| vis | stringArray | None | List of input measurement sets to flag |
| autocorr | bool | True | Flag autocorrelation data |
| shadow | bool | True | Flag shadowed antennas |
| scan | bool | True | Flag specified scans |
| scannumber | string | None | List of scans to be flagged |
| intents | string | POINTING, FOCUS, ATMOSPHERE, SIDE BAND | List of intents of scans to be flagged |
| edgespw | bool | True | Flag edge channels |
| fracspw | double | 0.0625 | Fraction of baseline correlator edge channels to be flagged |
| fracspwfps | double | 0.048387 | Fraction of ACA correlator edge channels to be flagged |
| online | bool | True | Apply the online flags |
| fileonline | string | None | File of online flags to be applied |
| template | bool | True | Apply a flagging template |
| filetemplate | stringArray | None | File that contains the flagging template |
| hm_tbuff | string | halfint | The time buffer computation heuristic |
| tbuff | any | [0.0,0.0] | List of time buffers (sec) to pad timerange in flag commands |
| qa0 | bool | True | QA0 flags |
| pipelinemode | string | automatic | The pipeline operating mode |
| flagbackup | bool | False | Backup preexisting flags before applying new ones |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

7.5 hifa_flagtargets

The `hifa_flagdata` data performs basic flagging operations on a list of measurements including:
o apply a flagging template

Task Description

Do science target flagging

The `hifa_flagtargets` task performs basic flagging operations on a list of science target measurement sets.

Keyword Arguments

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

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

template -- Apply flagging templates
default: True

filetemplate -- The name of an text file that contains the flagging template for issues with the science target data etc. If the template flags files is undefined a name of the form 'msname_flagtargetstemplate.txt' is assumed.
default: "

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

vis -- The list of input measurement sets. Defaults to the list of measurement defined in the pipeline context.
example:
default: "

flagbackup -- Back up any pre-existing flags.
default: False

--- pipeline task execution modes

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

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

Output

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

Examples

1. Do basic flagging on a science target measurement set
 hifa_flagtargets()

Parameter List

Table 42: hifa_flagtargets default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|-----------|--|
| vis | stringArray | None | List of input measurement sets to flag |
| template | bool | True | Apply a flagging template |
| filetemplate | stringArray | None | File that contains the flagging template |
| pipelinemode | string | automatic | The pipeline operating mode |
| flagbackup | bool | False | Backup preexisting flags before applying new ones |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

7.6 hifa_fluxcalflag

Search the builtin solar system flux calibrator line catalog for overlaps with the science spectral windows. Generate a list of line overlap regions and flagging commands.

Task Description

Locate and flag line regions in solar system flux calibrators
 Fills the model column with the model visibilities.

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

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

threshold -- If the fraction of an spw occupied by line regions is greater then threshold flag the entire spectral window.

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

vis -- The list of input measurement sets. Defaults to the list of measurment sets defined in the pipeline context.
 default: []
 example:

field -- The list of field names or field ids for which the models are to be set. Defaults to all fields with intent 'AMPLITUDE'.

default: "

example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to all data with amplitude intent.

default: "

example: 'AMPLITUDE'

spw -- The list of spectral windows and channels for which bandpasses are computed. Defaults to all science spectral windows.

default: "

example: '11,13,15,17'

--- pipeline task execution modes

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

default: True

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

default: True

Output

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

Examples

1. Locate known lines in any solar system object flux calibrators.

hifa_fluxcalflag()

Parameter List

Table 43: hifa_fluxcalflag default settings

| Parameter | type | default | description |
|---------------------|-------------|-----------|---------------------------------------|
| vis | stringArray | None | List of input measurement sets |
| field | string | None | List of field names or ids |
| intent | string | None | Observing intent of flux calibrators |
| spw | string | None | List of spectral window ids |
| pipelinemode | string | automatic | The pipeline operating mode |
| threshold | double | 0.75 | Threshold for flagging the entire spw |

| | | | |
|----------------------|--------|-------|---|
| appendlines | bool | False | Append user defined line regions to the line dictionary |
| linesfile | string | None | File containing user defined lines |
| applyflags | bool | True | Apply the computed flag commands |
| dryrun | bool | False | Run the task (False) or display the commands (True) |
| acceptresults | bool | True | Automatically accept results into the context |

7.7 hifa_fluxdb

Connect to the ALMA flux calibrator database

Task Description

Connect to flux calibrator database
Connect to the ALMA flux calibrator database

Keyword Arguments

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

vis -- List of input visibility files
default: none; example: vis='ngc5921.ms'

-- Pipeline task execution modes

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

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

Output

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

Example

1. Connect to the ALMA flux calibrator database
hifa_fluxdb()

Parameter List

Table 44: hifa_fluxdb default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|-----------|--|
| vis | stringArray | None | List of input measurement sets |
| pipelinemode | string | automatic | The pipeline operations mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically apply results to context |

7.8 hifa_gaincalsnr

The gaincal solution signal to noise is determined as follows.

- For each data set the list of source(s) to use for the per scan gaincal solution signal to noise estimation is compiled based on the values of the field, intent, and spw parameters.
- Source fluxes are determined for each spw and source combination.
- Fluxes in Jy are derived from the pipeline context.
- Pipeline context fluxes are derived from the online flux calibrator catalog, the ASDM, or the user via the flux.csv file.
- If no fluxes are available the task terminates.
- Atmospheric calibration and observations scans are determined for each spw and source combination.
- If intent is set to 'PHASE' are there are no atmospheric scans associated with the 'PHASE' calibrator, 'TARGET' atmospheric scans will be used in stead.
- If atmospheric scans cannot be associated with any of the spw and source combinations the task terminates.
- Science spws are mapped to atmospheric spws for each science spw and source combinations.
- If mappings cannot be determined for any of the spws the task terminates
- The median Tsys value for each atmospheric spw and source combination is determined from the SYSCAL table. Medians are computed first by channel, then by antenna, in order to reduce sensitivity to deviant values.
- The science spw parameters, exposure time(s), and integration time(s) are determined.
- The per scan sensitivity and signal to noise estimates are computed per science spectral window. Nominal Tsys and sensitivity values per receiver band provide by the ALMA project are used for this estimate.
- The QA score is based on how many signal to noise estimates greater then the requested singal to noise ration can be computed.

Task Description

Compute gaincal signal to noise ratios per spw

Compute the per scan gaincal solution signal to noise ratio per science spw

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

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

phasensr -- The required gaincal solution signal to noise
default: 25.0
example: phasepsnr = 20.0

bwedgefrac -- The fraction of the bandwidth edges that is flagged
default: 0.03125
example: bwedgefrac = 0.0

hm_nantennas -- The heuristics for determines the number of antennas to use in the signal to noise estimate. The options are 'all' and 'unflagged'. The 'unflagged' options is not currently supported.
default: 'all'
example: hm_nantennas='unflagged'

maxfracflagged -- The maximum fraction of an antenna that can be flagged before its is excluded from the signal to noise estimate.
default: 0.90
example: maxfracflagged=0.80

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context
default: ""
example: ['M82A.ms', 'M82B.ms']

field -- The list of field names of sources to be used for signal to noise estimation. Defaults to all fields with the standard intent.
default: ""
example: '3C279'

intent -- A string the intent against which the the selected fields are matched.
default: 'PHASE'
example: intent='BANDPASS'

spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all the science spectral windows for which there are both 'intent' and TARGET intents.
default: ""
example: '13,15'

--- pipeline task execution modes

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

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

Output

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

Examples

1. Estimate the per scan gaincal solution sensitivities and signal to noise ratios for all the science spectral windows.
hifa_gaincalsnr()

Parameter List

Table 45: hifa_gaincalsnr default settings

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

7.9 hifa_gfluxscale

Derive flux densities for point source transfer calibrators using flux models for reference calibrators.

Flux values are determined by:

- o computing complex gain phase only solutions for all the science spectral windows using the calibrator data selected by the 'reference' and 'refintent' parameters and the 'transfer' and 'transintent' parameters, and the value of the 'phaseupsolint' parameter.

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

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

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

Task Description

Derive flux density scales from standard calibrators

Derive flux densities for point source transfer calibrators using flux models for reference calibrators.

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

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

phaseupsolint -- Time solution intervals in CASA syntax for the phase solution.
default: 'inf'
example: 'inf', 'int', '100sec'

solint -- Time solution intervals in CASA syntax for the amplitude solution.
default: 'inf'
example: 'inf', 'int', '100sec'

minsnr -- Minimum signal to noise ratio for gain calibration solutions.
default: 2.0
example: 1.5, 0.0

hm_resolvedcals -- Heuristics method for handling resolved calibrators. The options are 'automatic' and 'manual'. In automatic mode antennas closer to the reference antenna than the uv distance where visibilities fall to 'peak_fraction' of the peak are used. In manual mode the antennas specified in 'antenna' are used.

antenna -- A comma delimited string specifying the antenna names or ids to be used for the fluxscale determination. Used in `hm_resolvedcals='manual'` mode.
default: ""
example: 'DV16,DV07,DA12,DA08'

peak_fraction -- The limiting UV distance from the reference antenna for antennas to be included in the flux calibration. Defined as the point where the calibrator visibilities have fallen to 'peak_fraction' of the peak value.

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context

default: ""

example: ['M32A.ms', 'M32B.ms']

reference -- A string containing a comma delimited list of field names defining the reference calibrators. Defaults to field names with intent '*AMP*'.

default: ""

example: 'M82,3C273'

transfer -- A string containing a comma delimited list of field names defining the transfer calibrators. Defaults to field names with intent '*PHASE*'.

default: ""

example: 'J1328+041,J1206+30'

refintent -- A string containing a comma delimited list of intents used to select the reference calibrators. Defaults to '*AMP*'.

default: ""

example: ", '*AMP*'

refspwmap -- Vector of spectral window ids enabling scaling across spectral windows. Defaults to no scaling

default: []

example: [1,1,3,3] (4 spws, reference fields in 1 and 3, transfer fields in 0,1,2,3)

reffile -- Path to a file containing flux densities for calibrators unknown to CASA. Values given in this file take precedence over the CASA-derived values for all calibrators except solar system calibrators. By default the path is set to the CSV file created by hifa_importdata, consisting of catalogue fluxes extracted from the ASDM and / or edited by the user.

default: ""

example: ", 'working/flux.csv'

transient -- A string containing a comma delimited list of intents defining the transfer calibrators. Defaults to '*PHASE*'.

default: ""

example: ", '*PHASE*'

refant -- A string specifying the reference antenna(s). By default this is read from the context.

default: ""

example: 'DV05'

--- pipeline task execution modes

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

default: False

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

Output

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

Issues

Should we add a spw window selection option here?

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

Examples

1. Compute flux flux values for the phase calibrator using model data from the amplitude calibrator.
hifa_gfluxscale ()

Parameter List

Table 46: hifa_gfluxscale default settings

| Parameter | Type | Default | Description |
|------------------------|----------|-----------|---|
| vis | string | None | List of input measurements sets |
| reference | variant | None | Reference calibrator field name(s) |
| transfer | variant | None | Transfer calibrator field name(s) |
| refintent | string | None | Observing intent of reference fields |
| transintent | string | None | Observing intent of transfer fields |
| refspwmap | intArray | None | Map accross spectral window boundaries |
| reffile | string | None | Path to file with fluxes for non-solar system calibrators |
| phaseupsolint | any | int | Phaseup correction solution interval |
| solint | any | inf | Amplitude correction solution interval |
| minsnr | double | 2.0 | Minimum SNR for gain solutions |
| refant | string | None | The name or ID of the reference antenna |
| hm_resolvedcals | string | automatic | The resolved calibrators heuristics method |
| antenna | string | None | Antennas to be used in fluxscale |
| peak_fraction | double | 0.2 | Fraction of peak visibility at uv-distance limit of antennas to be used |

| | | | |
|----------------------|--------|-----------|---|
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display commands (True) |
| acceptresults | bool | True | Automatically accept results into context |

7.10 hifa_importdata

Task Description

Imports data into the interferometry pipeline

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

Keyword Arguments

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

vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs, MSs, or tar files of MSs, If ASDM files are specified, they will be converted to MS format.

default: []

example: vis=['X227.ms', 'asdms.tar.gz']

session -- List of sessions to which the visibility files belong. Defaults to a single session containing all the visibility files, otherwise a session must be assigned to each vis file.

default: []

example: session=['session_1', 'session_2']

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

default: 'automatic'.

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

asis -- ASDM tables to convert as is

default: 'Antenna Station Receiver Source CalAtmosphere CalWVR'

example: 'Receiver', "

process_caldevice -- Ingest the ASDM caldevice table

default: False

example: True

overwrite -- Overwrite existing MSs on output.

default: False

bdf flags -- Apply BDF flags on line

default: True

ocorr_mode -- Read in cross- and auto-correlation data(ca), cross-correlation data oonly (co), or autocorrelation data only (ao).
default: ca

lazy -- Use the lazy filter import
default: False

dbservice -- Use online flux catalog on import
default: False

ocorr_mode -- Read in cross- and auto-correlation data(ca), cross-correlation data oonly (co), or autocorrelation data only (ao).
default: ca

createmms -- Create a multi-measurement set ('true') ready for parallel processing, or a standard measurement set ('false'). The default setting ('automatic') creates an MMS if running in a cluster environment.
default: automatic

clearcals -- Clear pipeline calibrations as needed by resetting the corrected data column to the data column and setting the model column to unity. For normal pipeline operations clearcals should be left on, and the pipeline will choose when it needs to reset the calibrations. In some reprocessing applications clearcals should be turned off.
default: True

--- pipeline task execution modes

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

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

Output

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

Examples

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

```
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

Table 47: hifa_importdata default settings

| Parameter | Type | Default | Description |
|--------------------------|-------------|---|---|
| vis | stringArray | None | List of input visibility data |
| session | stringArray | None | List of visibility data sessions |
| pipelinemode | string | automatic | The pipeline operating mode |
| asis | string | Antenna Station Receiver Source CalAtmosphere CalWVR | ASDM to convert as is |
| process_caldevice | bool | False | Import the caldevice table from the ASDM |
| overwrite | bool | False | Overwrite existing files on import |
| bdf flags | bool | True | Apply BDF flags on import |
| 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 | automatic | Create a MMS |
| clearcals | bool | True | Reinitialize pipeline calibrations as needed |
| dryrun | bool | False | Run the task (False) or display task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

7.11 hifa_linpolcal

Task Description

Compute polarization calibration
Compute a polarization calibration.

Keyword Arguments

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

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

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the context. CURRENTLY THE LIST MUST CONTAIN 1 MEASUREMENT SET.
default: ""
example: vis=['ngc5921a.ms', 'ngc5921b.ms', 'ngc5921c.ms']

--- pipeline task execution modes

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

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

Output

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

Parameter List

Table 48: hifa_linpolcal default settings

| Parameter | Type | Default | Description |
|-------------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets, '\\' for default |
| field | string | None | Set of data selection field names or ids |
| intent | string | None | Set of data selection intents |
| g0table | string | None | Name of table holding G0 gain - not accounting for source pol |
| delaytable | string | None | Name of table holding cross-hand delay |

| | | | |
|----------------------|--------|-----------|--|
| xyf0table | string | None | Name of table holding residual X-Y phase spectrum and source Q and U |
| g1table | string | None | Name of table holding G1 gain - accounting for source pol |
| df0table | string | None | Name of table holding instrument polarization gain |
| refant | string | None | Reference antenna names |
| spw | string | None | Set of data selection spectral window/channels |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results to the pipeline context |

7.12 hifa_spwphaseup

hif_spwphaseup performs tow functions

- o determines the spectral window mapping mode for the phase vs time calibrations and computes spectral window map that will be used to apply those calibrations
- o computes the per spectral window phase offset table that will be applied to the data to remove mean phase differences between the spectral windows

If hm_spwmapmode = 'auto' the spectral window map is computed using the following algorithm

- o estimate the per spectral window per scan signal to noise ratio of the phase calibrator observations
- o if the signal to noise of any single phase calibration spectral window is less than the value of 'phasesnr' hm_spwmapmode defaults to 'combine'
- o if all phase calibrator spectral windows meet the low signal to noise criterion then hm_spwmapmode defaults to default'
- o if the phase calibrator signal to noise values cannot be computed for any reason, for example there is no flux information, then hm_spwmapmode defaults to 'combine'

If hm_spwmapmode = 'combine' hifa_spwphaseup maps all the science windows to a single science spectral window. For example if the list of science spectral windows is [9, 11, 13, 15] then all the science spectral windows in the data will be combined and mapped to the science window 9 in the combined phase vs time calibration table.

If hm_spwmapmode = 'simple', a mapping from narrow science to wider science spectral windows is computed using the following algorithms:

- o construct a list of the bandwidths of all the science spectral windows
- o determine the maximum bandwidth in this list maxbandwidth
- o for each science spectral window with bandwidth less than maxbandwidth
- o construct a list of spectral windows with bandwidths greater than minfracmaxbw * maxbandwidth
- o select the spectral window in this list whose band center most closely matches the band center of the narr spectral window

- preferentially match within the same baseband if samebb is True

If `hm_spwmapmode = 'default'` the spw mapping is assumed to be one to one.

Phase offsets per spectral window are determined by computing a phase only gain calibration on the selected data, normally the high signal to noise bandpass calibrator observations, using the solution interval 'inf'.

At the end of the task the spectral window map and the phase offset calibration table in the pipeline are stored in the context for use by later tasks.

Task Description

Compute phase calibration spw map and per spw phase offsets
Compute the gain solutions.

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

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

hm_spwmapmode -- The spectral window mapping mode. The options are: 'auto', 'combine', 'simple', and 'default'. In 'auto' mode `hifa_spwphaseup` estimates the SNR of the phase calibrator observations and uses these estimates to choose between 'combine' mode (low SNR) and 'default' mode (high SNR). In combine mode all spectral windows are combined and mapped to one spectral window. In 'simple' mode narrow spectral windows are mapped to wider ones using an algorithm defined by 'maxnarrowbw', 'minfracmaxbw', and 'samebb'. In 'default' mode the spectral window map defaults to the standard one to one mapping.
default: 'auto'
example: `hm_spwmapmode='combine'`

maxnarrowbw -- The maximum bandwidth defining narrow spectral windows. Values must be in CASA compatible frequency units.
default: '300MHz'
example: `maxnarrowbw=""`

minfracmaxbw -- The minimum fraction of the maximum bandwidth in the set of spws to use for matching.
default: 0.8
example: `minfracmaxbw=0.75`

samebb -- Match within the same baseband if possible?
default: True
example: `samebb=False`

phasesnr -- The required gaincal solution signal to noise
default: 25.0
example: `phaseupsnr = 20.0`

bwedgefrac -- The fraction of the bandwidth edges that is flagged
default: 0.03125
example: bwedgefrac = 0.0

hm_nantennas -- The heuristics for determines the number of antennas to use in the signal to noise estimate. The options are 'all' and 'unflagged'. The 'unflagged' options is not currently supported.
default: 'all'
example: hm_nantennas='unflagged'

maxfracflagged -- The maximum fraction of an antenna that can be flagged before its is excluded from the signal to noise estimate.
default: 0.90
example: maxfracflagged=0.80

combine -- Data axes to combine for solving. Options are "", 'scan', 'spw', 'field' or any comma-separated combination.
default: ""
example: combine=""

minblperant -- Minimum number of baselines required per antenna for each solve. Antennas with fewer baaselines are excluded from solutions.
default: 4
example: minblperant=2

minsnr -- Solutions below this SNR are rejected.
default: 3.0

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context.
default: ""
example: ['M82A.ms', 'M82B.ms']

caltable -- The list of output calibration tables. Defaults to the standard pipeline naming convention.
default: ""
example: ['M82.gcal', 'M82B.gcal']

field -- The list of field names or field ids for which phase offset solutions are to be computed. Defaults to all fields with the default intent.
default: ""
example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to the BANDPASS observations.
default: ""
example: intent='PHASE'

spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all the science spectral windows.
default: ""

example: '13,15'

refant -- Reference antenna name(s) in priority order. Defaults to most recent values set in the pipeline context. If no reference antenna is defined in the pipeline context the CASA defaults are used.

default: ""

example: refant='DV01', refant='DV05,DV07'

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Examples

1. Compute the default spectral window map and the per spectral window phase offsets.

```
hif_spwphaseup()
```

2. Compute the default spectral window map and the per spectral window phase offsets set the spectral window mapping mode to 'simple'.

```
hif_spwphaseup(hm_spwmapmode='simple')
```

Parameter List

Table 49: hifa_spwphaseup default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|---------|---|
| vis | stringArray | None | List of input measurement sets |
| caltable | stringArray | None | List of output caltables |
| field | string | None | Set of data selection field names or ids |
| intent | string | None | Set of data selection observing intents |
| spw | string | None | Set of data selection spectral window/channels |
| hm_spwmapmode | string | auto | The spw mapping mode |
| maxnarrowbw | string | 300MHz | The maximum bandwidth defining narrow spectral windows |
| minfracmaxbw | double | 0.8 | The minimum fraction of the maximum bandpass for spw matching |

| | | | |
|-----------------------|--------|-----------|---|
| samebb | bool | True | Match within the same baseband if possible? |
| phasesnr | double | 25.0 | The minimum snr for triggering spw combination in auto spw mapping mode |
| bwedgefrac | double | 0.03125 | The fraction of the bandwidth edge that is flagged |
| hm_nantennas | string | all | The antenna selection heuristic |
| maxfracflagged | double | 0.90 | The maximum fraction of data flagged per antenna |
| combine | string | None | Data axes which to combine for solve (scan, spw, and/or field) |
| refant | string | None | Reference antenna names |
| minblperant | int | 4 | Minimum baselines per antenna required for solve |
| minsnr | double | 3.0 | Reject solutions below this SNR |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run task (False) or display the command (True) |
| acceptresults | bool | True | Automatically accept results into the context |

7.13 hifa_timegaincal

The complex gains are derived from the data column (raw data) divided by the model column (usually set with hif_setjy). The gains are obtained for the specified solution intervals, spw combination and field combination. One gain solution is computed for the science targets and one for the calibrator targets.

Good candidate reference antennas can be determined using the hif_refant task.

Previous calibrations that have been stored in the pipeline context are applied on the fly. Users can interact with these calibrations via the hif_export_calstate and hif_import_calstate tasks.

Task Description

Determine temporal gains from calibrator observations
 Compute the gain solutions.

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

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

calsolint -- Time solution interval in CASA syntax for calibrator source solutions.
 default: 'int'

example: 'inf', 'int', '100sec'

targetsolint -- Time solution interval in CASA syntax for target source solutions.

default: 'inf'

example: 'inf', 'int', '100sec'

combine -- Data axes to combine for solving. Options are ", 'scan', 'spw', 'field' or any comma-separated combination.

default: ""

example: combine=""

minblperant -- Minimum number of baselines required per antenna for each solve. Antennas with fewer baselines are excluded from solutions.

default: 4

example: minblperant=2

calminsnr -- Solutions below this SNR are rejected for calibrator solutions.

default: 2.0

targetminsnr -- Solutions below this SNR are rejected for science target solutions.

default: 3.0

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets specified in the pipeline context

default: ""

example: ['M82A.ms', 'M82B.ms']

calamptable -- The list of output diagnostic calibration amplitude tables for the calibration targets. Defaults to the standard pipeline naming convention.

default: ""

example: ['M82.gcal', 'M82B.gcal']

calphasetable -- The list of output calibration phase tables for the calibration targets. Defaults to the standard pipeline naming convention.

default: ""

example: ['M82.gcal', 'M82B.gcal']

amptable -- The list of output calibration amplitude tables for the calibration and science targets. Defaults to the standard pipeline naming convention.

default: ""

example: ['M82.gcal', 'M82B.gcal']

targetphasetable -- The list of output phase calibration tables for the science targets. Defaults to the standard pipeline naming convention.

default: ""

example: ['M82.gcal', 'M82B.gcal']

field -- The list of field names or field ids for which gain solutions are to be computed. Defaults to all fields with the standard intent.

default: ""

example: '3C279', '3C279, M82'

intent -- A string containing a comma delimited list of intents against which the the selected fields are matched. Defaults to the equivalent of 'AMPLITUDE,PHASE,BANDPASS'.

default: "

example: ", 'PHASE'

spw -- The list of spectral windows and channels for which gain solutions are computed. Defaults to all science spectral windows.

default: "

example: '3C279', '3C279, M82'

smodel -- Point source Stokes parameters for source model (experimental) Defaults to using standard MODEL_DATA column data.

default: []

example: [1,0,0,0] (I=1, unpolarized)

refant -- Reference antenna name(s) in priority order. Defaults to most recent values set in the pipeline context. If no reference antenna is defined in the pipeline context use the CASA defaults.

default: "

example: refant='DV01', refant='DV05,DV07'

solnorm -- Normalise the gain solutions

default: False

--- pipeline task execution modes

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

default: False

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

default: True

Output

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

Examples

1. Compute standard per scan gain solutions that will be used to calibrate the target.

hifa_timegaincal()

Parameter List

Table 50: hifa_timegaincal default settings

| Parameter | Type | Default | Description |
|-----------|-------------|---------|--------------------------------|
| vis | stringArray | None | List of input measurement sets |

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

7.14 hifa_tsyscal

Task Description

Derive a Tsys calibration table
Derive the Tsys calibration for list of measurement sets

Keyword Arguments

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

default: 'automatic'.

chantol -- The tolerance in channels for mapping atmospheric calibration windows (TDM) to science windows (FDM or TDM)

default: 1

example: 5

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

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

vis -- List of input visibility files

default: none; example: vis='ngc5921.ms'

caltable -- Name of output gain calibration tables

default: none; example: caltable='ngc5921.gcal'

-- Pipeline task execution modes

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

default: True

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

default: True

Output

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

Example

1. Compute the tsys calibration tables for a list of measurement sets

hif_tsyscal()

Parameter List

Table 51: hifa_tsyscal default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|-----------|--|
| vis | stringArray | None | List of input measurement sets |
| caltable | stringArray | None | List of output caltable(s) |
| chantol | int | 1 | Tsys spectral window map channel tolerance |
| pipelinemode | string | automatic | The pipeline operations mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically apply results to context |

7.15 hifa_tsysflag

Task Description

Flag deviant system temperature measurements

hif_tsysflag tries to flag all deviant system temperature measurements in the system temperature calibration table. It does this by running a sequence of flagging tests, each designed to look for a different type of error. The tests are:

1. Flag Tsys spectra with high median values
2. Flag Tsys spectra with high median derivatives. This is meant to spot spectra that are 'ringing'.
3. Flag the edge channels of the Tsys spectra in each SpW.
4. Flag Tsys spectra whose shape is different from that associated with the BANDPASS intent.
5. Flag 'birdies'.
6. Flag the Tsys spectra of all antennas in a timestamp and spw if proportion of antennas already flagged in this timestamp and spw exceeds a threshold, and flag Tsys spectra for all antennas and all timestamps in a spw, if proportion of antennas that are already entirely flagged in all timestamps exceeds a threshold.

Keyword Arguments

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

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

flag_nmedian -- True to flag Tsys spectra with high median value.
default: True

fnm_limit -- Flag spectra with median value higher than fnm_limit * median of this measure over all spectra.
default: 2.0

fnm_byfield -- Evaluate the nmedian metric separately for each field.
default: False

flag_derivative -- True to flag Tsys spectra with high median derivative.
default: True

fd_max_limit -- Flag spectra with median derivative higher than fd_max_limit * median of this measure over all spectra.
default: 5.0

flag_edgechans -- True to flag edges of Tsys spectra.
default: True

fe_edge_limit -- Flag channels whose channel to channel difference > fe_edge_limit * median across spectrum.
default: 3.0

flag_fieldshape -- True to flag Tsys spectra with a radically different shape to those of the ff_refintent.
default: True

ff_refintent -- Data intent that provides the reference shape for 'flag_fieldshape'.
default: BANDPASS

ff_max_limit -- Flag Tsys spectra with 'fieldshape' metric values > ff_max_limit.
default: 5.0

flag_birdies -- True to flag channels covering sharp spectral features.
default: True

fb_sharps_limit -- Flag channels bracketing a channel to channel difference > fb_sharps_limit.
default: 0.05

flag_toomany -- True to flag Tsys spectra for which a proportion of antennas for given timestamp and/or proportion of antennas that are entirely flagged in all timestamps exceeds their respective thresholds.
default: True

tmf1_limit -- Flag Tsys spectra for all antennas in a timestamp and spw if proportion of antennas already flagged in this timestamp and spw exceeds tmf1_limit.
default: 0.666

tmef1_limit -- Flag Tsys spectra for all antennas and all timestamps in a spw, if proportion of antennas that are already entirely flagged in all timestamps exceeds tmef1_limit.
default: 0.666

metric_order -- Order in which to evaluate the flagging metrics that are enables. Disabled metrics are skipped.
default: 'nmedian,derivative,edgechans,fieldshape,birdies,toomany'

normalize_tsys -- True to create a normalized Tsys table that is used to evaluate the Tsys flagging metrics. All newly found flags are also applied to the original Tsys caltable that continues to be used for subsequent calibration.
default: False

--- Pipeline context defined parameter arguments which can be set only in 'interactive mode'

caltable -- List of input Tsys calibration tables
default: [] - Use the table currently stored in the pipeline context.
example: caltable=['X132.ms.tsys.s2.tbl']

--- Pipeline task execution modes

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

default: True

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

Output

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

Examples

1. Flag Tsys measurements using currently recommended tests:

```
hif_tsysflag()
```

2. Flag Tsys measurements using all recommended tests apart from that using the 'fieldshape' metric.

```
hif_tsysflag(flag_fieldshape=False)
```

Parameter List

Table 52: hifa_tsysflag default settings

| Parameter | Type | Default | Description |
|------------------------|-------------|----------|--|
| vis | stringArray | None | List of input measurement sets (Not used) |
| caltable | stringArray | None | List of input caltables |
| flag_nmedian | bool | True | True to flag Tsys spectra with high median value |
| fnm_limit | double | 2.0 | Flag spectra with median greater than $fnm_limit * median$ over all spectra |
| fnm_byfield | bool | False | Evaluate the nmedian metric separately for each field. |
| flag_derivative | bool | True | True to flag Tsys spectra with high median derivative |
| fd_max_limit | double | 5.0 | Flag spectra with median derivative higher than $fd_max_limit * median$ of this measure over all spectra |
| flag_edgechans | bool | True | True to flag edges of Tsys spectra |
| fe_edge_limit | double | 3.0 | Flag channels whose channel to channel difference greater than $fe_edge_limit * median$ across spectrum |
| flag_fieldshape | bool | True | True to flag Tsys spectra with a radically different shape to those of the <code>ff_refintent</code> |
| ff_refintent | string | BANDPASS | Data intent providing the reference shape for <code>'flag_fieldshape'</code> |
| ff_max_limit | double | 5.0 | Flag Tsys spectra with <code>'fieldshape'</code> metric greater than <code>ff_max_limit</code> |
| flag_birdies | bool | True | True to flag channels covering sharp spectral features |

| | | | |
|------------------------|--------|---|---|
| fb_sharps_limit | double | 0.05 | Flag channels bracketing a channel to channel difference greater than fb_sharps_limit |
| flag_toomany | bool | True | True to flag Tsys spectra for which a proportion of timestamps or proportion of antennas that are entirely flagged exceeds their respective thresholds. |
| tmf1_limit | double | 0.666 | Flag all Tsys spectra within a timestamp for an antenna if proportion flagged already exceeds tmf1_limit |
| tmef1_limit | double | 0.666 | Flag all Tsys spectra for all antennas in a spw, if proportion of antennas that are already entirely flagged in all timestamps exceeds tmef1_limit |
| metric_order | string | nmedian, derivative, edgechans, fieldshape, birdies, toomany | Order in which to evaluate the flagging metric(s); inactive metrics are skipped. |
| normalize_tsys | bool | False | Normalize Tsys prior to computing the flagging metric(s) |
| pipelinemode | string | automatic | The pipeline operations mode |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | Bool | True | Automatically apply results to context |

7.16 hifa_wvrgcalflag

Task Description

First, generate a gain table based on the Water Vapour Radiometer data in each vis file.

Second, apply the wvr calibration to the data specified by 'flag_intent', calculate flagging 'views' showing the ratio phase-rms with wvr / phase-rms without wvr for each scan. A ratio < 1 implies that the phase noise is improved, a score > 1 implies that it is made worse.

Third, search the flagging views for antennas with anomalous high values. If any are found then recalculate the wvr calibration with the 'wvrflag' parameter set to ignore their data and interpolate results from other antennas according to 'maxdism' and 'minnumants'.

Fourth, if the overall QA score for the final wvr correction of a vis file is greater than the value in 'accept_threshold' then make available the wvr calibration file for merging into the context and use in the subsequent reduction.

vis -- List of input visibility files

default: none, in which case the vis files to be used will be read from the context.

example: vis=['ngc5921.ms']

caltable -- List of output gain calibration tables

default: none, in which case the names of the caltables will be generated automatically.

example: caltable='ngc5921.wvr'

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

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

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

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

tie -- Use the same atmospheric phase correction coefficients when calculating the wvr correction for all sources in the 'tie'. If 'tie' is not empty then 'segsource' is forced to be True. Ignored unless hm_tie='manual'.
default: []
example: ['3C273,NGC253', 'IC433,3C279']

sourceflag -- Flag the WVR data for these source(s) as bad and do not produce corrections for it. Requires segsource=True
default: []
example: ['3C273']

nsol -- Number of solutions for phase correction coefficients during this observation, evenly distributed in time throughout the observation. It is used only if segsource=False because if segsource=True then the coefficients are recomputed whenever the telescope moves to a new source (within the limits imposed by 'tie').
default: 1

disperse -- Apply correction for dispersion
default: False

wvrflag -- Flag the WVR data for these antenna(s) as bad and replace its data with interpolated values
default: []
example: ['DV03','DA05','PM02']

hm_smooth -- If 'manual' set the 'smooth' parameter to the user-specified value. If 'automatic', run the wvrgcal task with the range of 'smooth' parameters required to match the integration time of the wvr data to that of the interferometric data in each spectral window.

smooth -- Smooth WVR data on this timescale before calculating the correction. Ignored unless hm_smooth='manual'.
default: ''

scale -- Scale the entire phase correction by this factor.

default: 1

maxdistrm -- Maximum distance in meters of an antenna used for interpolation from a flagged antenna.

default: 500

example: 550

minnumants -- Minimum number of nearby antennas (up to 3) used for interpolation from a flagged antenna.

default: 2

example: 3

mingoodfrac -- Minimum fraction of good data per antenna

default: 0.8

example: 0.7

refant -- Ranked comma delimited list of reference antennas

default: "

example: 'DV02,DV06'

flag_intent -- The data intent(s) on whose wvr correction results the search for bad wvr antennas is to be based.

A 'flagging view' will be calculated for each specified intent, in each spectral window in each vis file.

Each 'flagging view' will consist of a 2-d image with dimensions ['ANTENNA', 'TIME'], showing the phase noise after the wvr correction has been applied.

If flag_intent is left blank, the default, the flagging views will be derived from data with the default bandpass calibration intent i.e. the first in the list BANDPASS, PHASE, AMPLITUDE for which the measurement set has data.

default "

qa_intent -- The list of data intents on which the wvr correction is to be tried as a means of estimating its effectiveness.

A QA 'view' will be calculated for each specified intent, in each spectral window in each vis file.

Each QA 'view' will consist of a pair of 2-d images with dimensions ['ANTENNA', 'TIME'], one showing the data phase-noise before the wvr application, the second showing the phase noise after (both 'before' and 'after' images have a bandpass calibration applied as well).

An overall QA score is calculated for each vis file, by dividing the 'before' images by the 'after' and taking the median of the result. An overall score of 1 would correspond to no change in the phase noise, a score > 1 implies an improvement.

If the overall score for a vis file is less than the value in 'accept_threshold' then the wvr calibration file is not made available for merging into the context for use in the subsequent reduction.

default: 'BANDPASS,PHASE'

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

lacks BANDPASS data.
 default: "

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

Example

1. Compute the WVR calibration for all the measurement sets.
 hifa_wvrgcalflag (hm_tie='automatic')

Parameter List

Table 53: hifa_wvrgcalflag default settings

| Parameter | Type | Default | Description |
|--------------------|-------------|-----------|---|
| vis | stringArray | None | List of input visibility files |
| caltable | stringArray | None | List of output gain calibration tables |
| hm_toffset | string | automatic | Toffset computation heuristic method |
| toffset | double | 0 | Time offset (sec) between IF and WVR data |
| segsource | bool | True | Compute new coefficient calculation for each source |
| sourceflag | stringArray | None | Flag the WVR data for these source(s) |
| hm_tie | string | automatic | Tie computation heuristics method |
| tie | stringArray | None | Sources for which to use the same atmospheric phase correction coefficients |
| nsol | int | 1 | Number of solutions for phase correction coefficients |
| disperse | bool | False | Apply correction for dispersion |
| wvrflag | stringArray | None | Flag the WVR data for these antenna(s) replace with interpolated values |
| hm_smooth | string | automatic | Smoothing computation heuristics method |
| smooth | string | None | Smooth WVR data on the given timescale before calculating the correction |
| scale | double | 1. | Scale the entire phase correction by this factor |
| maxdistm | double | 500. | Maximum distance (m) of an antenna used for interpolation for a flagged antenna |
| minnumants | int | 2 | Minimum number of near antennas (up to 3) required for interpolation |
| mingoodfrac | double | 0.8 | Minimum fraction of good data per antenna |
| refant | string | None | Ranked list of reference antennas |

| | | | |
|---------------------------|--------|--------------------|--|
| flag_intent | string | None | Data intents to use in detecting and flagging bad wvr antennas |
| qa_intent | string | BANDPASS, PHASE | Data intents to use in estimating the effectiveness of the wvr correction |
| qa_bandpass_intent | string | None | Data intent to use for the bandpass calibration in the qa calculation |
| accept_threshold | double | 1.0 | Improvement ratio (phase-rms without wvr / phase-rms with wvr) above which wvr calibration file will be accepted |
| flag_hi | bool | True | True to flag high figure of merit outliers |
| fhi_limit | double | 10.0 | Flag figure of merit values higher than limit * MAD |
| fhi_minsample | int | 5 | Minimum number of samples for valid MAD estimate |
| 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 |

7.17 hifa_wvrgcal

Task Description

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

vis -- List of input visibility files

default: none, in which case the vis files to be used will be read from the context.

example: vis=['ngc5921.ms']

caltable -- List of output gain calibration tables

default: none, in which case the names of the caltables will be generated automatically.

example: caltable='ngc5921.wvr'

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

default: 'automatic'

toffset -- Time offset (sec) between interferometric and WVR data

default: 0

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

default: True

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

tie -- Use the same atmospheric phase correction coefficients when calculating the wvr correction for all sources in the 'tie'. If 'tie' is not empty then 'segsource' is forced to be True. Ignored unless hm_tie='manual'.
default: []
example: ['3C273,NGC253', 'IC433,3C279']

sourceflag -- Flag the WVR data for these source(s) as bad and do not produce corrections for it. Requires segsource=True.
default: []
example: ['3C273']

nsol -- Number of solutions for phase correction coefficients during this observation, evenly distributed in time throughout the observation. It is used only if segsource=False because if segsource=True then the coefficients are recomputed whenever the telescope moves to a new source (within the limits imposed by 'tie').
default: 1

disperse -- Apply correction for dispersion
default: False

wvrflag -- Flag the WVR data for the listed antennas as bad and replace their data with values interpolated from the 3 nearest antennas with unflagged data.
default: []
example: ['DV03', 'DA05', 'PM02']

hm_smooth -- If 'manual' set the 'smooth' parameter to the user-specified value. If 'automatic', run the wvrgcal task with the range of 'smooth' parameters required to match the integration time of the wvr data to that of the interferometric data in each spectral window.

smooth -- Smooth WVR data on this timescale before calculating the correction. Ignored unless hm_smooth='manual'.
default: "

scale -- Scale the entire phase correction by this factor.
default: 1

maxdistm -- Maximum distance in meters of an antenna used for interpolation from a flagged antenna.
default: 500
example: 550

minnumants -- Minimum number of nearby antennas (up to 3) used for interpolation from a flagged antenna.
default: 2
example: 3

mingoodfrac -- Minimum fraction of good data per antenna
default: 0.8

refant -- Ranked comma delimited list of reference antennas.
default: "
example 'DV01,DV02'

qa_intent -- The list of data intents on which the wvr correction is to be tried as a means of estimating its effectiveness.

A QA 'view' will be calculated for each specified intent, in each spectral window in each vis file.

Each QA 'view' will consist of a pair of 2-d images with dimensions ['ANTENNA', 'TIME'], one showing the data phase-noise before the wvr application, the second showing the phase noise after (both 'before' and 'after' images have a bandpass calibration applied as well).

An overall QA score is calculated for each vis file, by dividing the 'before' images by the 'after' and taking the median of the result. An overall score of 1 would correspond to no change in the phase noise, a score > 1 implies an improvement.

If the overall score for a vis file is less than the value in 'accept_threshold' then the wvr calibration file is not made available for merging into the context for use in the subsequent reduction.

If you do not want any QA calculations then set qa_intent="".
default: "
example: 'PHASE'

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

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

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

Example

1. Compute the WVR calibration for all the measurement sets.
hifa_wvrgcal (hm_tie='automatic')

Parameter List

Table 54: hifa_wvrgcal default settings

| Parameter | Type | Default | Description |
|---------------------------|-------------|-----------|--|
| vis | stringArray | None | List of input visibility files |
| caltable | stringArray | None | List of output gain calibration tables |
| hm_toffset | string | automatic | Toffset computation heuristic method |
| toffset | double | 0 | Time offset (sec) between IF and WVR data |
| segsource | bool | True | Compute new coefficient calculation for each source |
| sourceflag | stringArray | None | Flag the WVR data for these source(s) |
| hm_tie | string | automatic | Tie computation heuristics method |
| tie | stringArray | None | Sources for which to use the same atmospheric phase correction coefficients |
| nsol | int | 1 | Number of solutions for phase correction coefficients |
| disperse | bool | False | Apply correction for dispersion |
| wvrflag | stringArray | None | Flag the WVR data for these antenna(s) replace with interpolated values |
| hm_smooth | string | automatic | Smoothing computation heuristics method |
| smooth | string | None | Smooth WVR data on the given timescale before calculating the correction |
| scale | double | 1. | Scale the entire phase correction by this factor |
| maxdistm | double | 500. | Maximum distance (m) of an antenna used for interpolation for a flagged antenna |
| minnumants | int | 2 | Minimum number of near antennas (up to 3) required for interpolation |
| mingoodfrac | double | 0.8 | Minimum fraction of good data per antenna |
| refant | string | None | Ranked list of reference antennas |
| qa_intent | string | None | Data intents to use in estimating the effectiveness of the wvr correction |
| qa_bandpass_intent | string | None | Data intent to use for the bandpass calibration in the qa calculation |
| qa_spw | string | None | Data spw(s) to use in estimating the effectiveness of the wvr correction |
| accept_threshold | double | 1.0 | Improvement ratio (phase-rms without wvr / phase-rms with wvr) above which wvr calibration file will be accepted |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results to the pipeline context |

8 Single-Dish Task Descriptions

8.1 hsd_applycal

hif_applycal applies the precomputed calibration tables stored in the pipeline context to the set of visibility files using predetermined field and spectral window maps and default values for the interpolation schemes. Users can interact with the pipeline calibration state using the tasks hif_export_calstate and hif_import_calstate.

Task Description

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

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

applymode -- Calibration apply mode="'calflagstrict': calibrate data and apply flags from solutions using the strict flagging convention

'trial': report on flags from solutions, dataset entirely unchanged

'flagonly': apply flags from solutions only, data not calibrated

'calonly': calibrate data only, flags from solutions NOT applied

'calflagstrict':

'flagonlystrict': same as above except flag spws for which calibration is unavailable in one or more tables (instead of allowing them to pass uncalibrated and unflagged)

default: "

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

default: 'automatic'.

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

vis -- The list of input measurement sets. Defaults to the list of measurement sets in the pipeline context.

default: []

example: ['X227.ms']

field -- A string containing the list of field names or field ids to which the calibration will be applied. Defaults to all fields in the pipeline context.

default: "

example: '3C279', '3C279, M82'

intent -- A string containing a the list of intents against which the selected fields will be matched. Defaults to all supported intents in the pipeline context.

default: "
example: '*TARGET*'

spw -- The list of spectral windows and channels to which the calibration will be applied. Defaults to all science windows in the pipeline context.

default: "
example: '17', '11, 15'

antenna -- The list of antennas to which the calibration will be applied. Defaults to all antennas. Not currently supported.

--- pipeline task execution modes

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

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

Output

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

Issues

There is some discussion about the appropriate values of calwt. Given properly scaled data, the correct value should be the CASA default of True. However at the current time ALMA is suggesting that calwt be set to True for applying observatory calibrations, e.g. antenna positions, WVR, and system temperature corrections, and to False for applying instrument calibrations, e.g. bandpass, gain, and flux.

Examples

1. Apply the calibration to the target data
hsd_applycal (intent='TARGET')

Parameter List

Table 55: hsd_applycal default settings

| Parameter | Type | Default | Description |
|-----------|-------------|---------|--|
| vis | stringArray | None | List of input measurement sets |
| field | string | None | Set of data selection field names or ids |
| intent | string | None | Set of data selection observing intents |
| spw | string | None | Set of data selection spectral window/channels |
| antenna | string | None | Set of data selection antenna ids |

| | | | |
|----------------------|-----------|-----------|---|
| applymode | string | None | Calibration mode: ""="calflagstrict", "calflag", "calflagstrict", "trial", "flagonly", "flagonlystrict", or "calonly" |
| calwt | boolArray | True | Calibrate the weights as well as the data |
| flagbackup | bool | True | Backup the flags before the apply |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run task (False) or display the command (True) |
| acceptresults | bool | True | Automatically accept results into the context |

8.2 hsd_baseline

Task Description

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

The hsd_baseline task subtracts baseline from calibrated spectra. By default, the task tries to find spectral line feature using line detection and validation algorithms. Then, the task puts a mask on detected lines and perform baseline subtraction. The user is able to turn off automatic line masking by setting linewidth parameter, which specifies pre-defined line window.

Fitting order is automatically determined by default. It can be disabled by specifying fitorder as non-negative value. In this case, the value specified by fitorder will be used.

Keyword Arguments

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

fitfunc -- fitting function for baseline subtraction. You can only choose cubic spline ('spline' or 'cspline')
default: 'cspline'.

fitorder -- Fitting order for polynomial. For cubic spline, it is used to determine how much the spectrum is segmented into. Default (-1) is to determine the order automatically.
default: -1 (auto determination)

linewidth -- 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. Two types of format are acceptable. One is channel-based specification, [min_chan,max_chan] and the other is specification by central frequency in GHz plus line range as velocity in km/s, [center_freq, min_vel, max_vel] You can set multiple line windows by using nested list. However, the above notation must not be mixed in the list for multiple line windows.
default: [] (do line detection and validation)
example: [100.0,200.0], [115.0,-10.0,10.0]

edge -- number of edge channels to be dropped from baseline subtraction. The value must be a list with length of 2, whose values specifies left and right edge channels respectively.
default: [] ([0,0])

example: [10,10]

broadline -- Try to detect broad component of spectral line if True.
default: True

clusteringalgorithm -- selection of the algorithm used in the clustering analysis to check the validity of detected line features. 'kmean' algorithm and hierarchical clustering algorithm 'hierarchy' are so far implemented.
default: 'kmean'.

deviationmask -- Apply deviation mask in addition to masks determined by the automatic line detection.
default: True

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

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

infiles -- List of data files. These must be a name of Scantables that are registered to context via hsd_importdata task.
default: []
example: vis=['X227.PM01.asap', 'X227.PM02.asap']

field -- Data selection by field.
default: " (all fields)
example: '1' (select by FIELD_ID) 'M100*' (select by field name)

antenna -- Data selection by antenna.
default: " (all antennas)
example: '1' (select by ANTENNA_ID) 'PM03' (select by antenna name)

spw -- Data selection by spw.
default: " (all spws)
example: '3,4' (generate caltable for spw 3 and 4) ['0','2'] (spw 0 for first data, 2 for second)

pol -- Data selection by pol.
default: " (all polarizations)
example: '0' (generate caltable for pol 0) ['0~1','0'] (pol 0 and 1 for first data, only 0 for second)

--- **pipeline task execution modes**

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

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

Output

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

Parameter List

Table 56: hsd_baseline default settings

| Parameter | Type | Default | Description |
|----------------------------|-------------|-----------|--|
| fitfunc | string | cspline | Fitting function for baseline subtraction |
| fitorder | int | -1 | Fitting order for baseline subtraction |
| linewindow | intArray | None | Pre-defined line window |
| edge | intArray | None | Edge channels to be dropped |
| broadline | bool | True | Try to detect broad component of the line |
| clusteringalgorithm | string | kmean | Algorithm for line validation clustering algorithm |
| deviationmask | bool | True | Apply deviation mask in addition to detected line masks |
| pipelinemode | string | automatic | The pipeline operating mode |
| infile | stringArray | None | List of input files to be calibrated (default all) |
| field | string | None | Select data by field |
| antenna | string | None | Select data by antenna |
| spw | any | None | Select data by IF IDs (spectral windows), e.g. '3,5,7' ('\='all) |
| pol | any | None | Select data by polarizations, e.g. '0~1' ('\='all) |
| dryrun | bool | False | Run the task (False) or display task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

8.3 hsd_bflag

The hsd_bflag 6. Flagging Data are flagged based on several flagging rules. Available rules are: expected rms, calculated rms, and running mean of both pre-fit and post-fit spectra.

Tsys flagging is also available. In addition, the heuristics script creates many plots for each stage. Those plots are included in the weblog.

Task Description

Flag spectra based on predefined criteria of single-dish pipeline

Keyword Arguments

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

iteration -- Number of iterations to perform sigma clipping to calculate threshold value of flagging.
default: 5

edge -- Number of channels to be dropped from the edge. The value must be a list of integer with length of one or two. If list length is one, same number will be applied both side of the band.
default: [0,0]
example: [10,20], [10]

flag_tsys -- Activate (True) or deactivate (False) Tsys flag.
default: True

tsys_thresh -- Threshold value for Tsys flag.
default: 3.0

flag_weath -- Activate (True) or deactivate (False) weather flag. Since weather flagging is not implemented yet. Setting True has no effect at the moment.
default: False

weath_thresh -- Threshold value for weather flag.
default: 3.0

flag_prfre -- Activate (True) or deactivate (False) flag by expected rms of pre-fit spectra.
default: True

prfre_thresh -- Threshold value for flag by expected rms of pre-fit spectra.
default: 3.0

flag_pofre -- Activate (True) or deactivate (False) flag by expected rms of post-fit spectra.
default: True

pofre_thresh -- Threshold value for flag by expected rms of post-fit spectra.
default: 1.3333

flag_prfr -- Activate (True) or deactivate (False) flag by rms of pre-fit spectra.
default: True

prfr_thresh -- Threshold value for flag by rms of pre-fit spectra.
default: 4.5

flag_pofr -- Activate (True) or deactivate (False) flag by rms of post-fit spectra.
default: True

pofr_thresh -- Threshold value for flag by rms of post-fit spectra.
default: 4.0

flag_prfrm -- Activate (True) or deactivate (False) flag by running mean of pre-fit spectra.
default: True

prfrm_thresh -- Threshold value for flag by running mean of pre-fit spectra.
default: 5.5

prfrm_nmean -- Number of channels for running mean of pre-fit spectra.
default: 5

flag_pofrm -- Activate (True) or deactivate (False) flag by running mean of post-fit spectra.
default: True

pofrm_thresh -- Threshold value for flag by running mean of post-fit spectra.
default: 5.0

pofrm_nmean -- Number of channels for running mean of post-fit spectra.
default: 5

flag_user -- Activate (True) or deactivate (False) user-defined flag. Since user flagging is not implemented yet. Setting True has no effect at the moment.
default: False

user_thresh -- Threshold value for flag by user-defined rule.
default: 3.0

plotflag -- True to plot result of data flagging.
default: True

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

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

infile -- ASDM or MS files to be processed. This parameter behaves as data selection parameter. The name specified by infile must be registered to context before you run hsd_bflag.
default: [] (process all data in context)

antenna -- Data selection by antenna names or ids.
default: " (all antennas)
example: 'PM03,PM04'

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

spw -- Data selection by spw IDs.
default: " (all spws)
example: '3,4' (spw 3 and 4)

pol -- Data selection by pol.
default: " (all polarizations)

example: 'XX,YY' (correlation XX and YY)

--- pipeline task execution modes

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

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

Output

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

Parameter List

Table 57: hsd_bflag default settings

| Parameter | Type | Default | Description |
|---------------------|----------|---------|---|
| iteration | int | 5 | Number of iterations to perform sigma clipping to calculate threshold |
| edge | intArray | 0,0 | Number of edge channel to be excluded from statistic calculation to flag data |
| flag_tsys | bool | True | Flag data by Tsys value |
| tsys_thresh | double | 3.0 | Threshold for Tsys flag |
| flag_weath | bool | False | Flag data by weather (not implemented yet) |
| weath_thresh | double | 3.0 | Threshold for weather flag |
| flag_prfre | bool | True | Flag data by EXPECTED RMS of pre-fit spectra |
| prfre_thresh | double | 3.0 | Threshold for EXPECTED RMS of pre-fit spectra flag |
| flag_pofre | bool | True | Flag data by EXPECTED RMS of post-fit spectra |
| pofre_thresh | double | 1.3333 | Threshold for EXPECTED RMS of post-fit spectra flag |
| flag_prfr | bool | True | Flag data by RMS of pre-fit spectra |
| prfr_thresh | double | 4.5 | Threshold for RMS of pre-fit flag |
| flag_pofr | bool | True | Flag data by RMS of post-fit spectra |
| pofr_thresh | double | 4.0 | Threshold for RMS of post-fit spectra flag |
| flag_prfrm | bool | True | Flag data by running mean of pre-fit spectra |
| prfrm_thresh | double | 5.5 | Threshold for running mean of pre-fit spectra flag |
| prfrm_nmean | int | 5 | Number of channel for running mean of pre-fit spectra flag |

| | | | |
|----------------------|-------------|-----------|--|
| flag_pofrm | bool | True | Flag data by running mean of post-fit spectra |
| pofrm_thresh | double | 5.0 | Threshold for running mean of post-fit spectra flag |
| pofrm_nmean | int | 5 | Number of channel for running mean of post-fit spectra flag |
| flag_user | bool | False | Flag data by user flag (not implemented yet) |
| user_thresh | double | 5.0 | Threshold for user flag |
| plotflag | bool | True | Create plots for flagging |
| pipelinemode | string | automatic | The pipeline operating mode |
| infile | stringArray | None | List of input files to be flagged ('\='=all) |
| antenna | string | None | Select data by antenna names or ids, e.g. 'PM03,PM04' ('\='=all) |
| field | string | None | Select data by field names or ids, e.g. 'M100,Sgr*' ('\='=all) |
| spw | string | None | Select data by spectral windows, e.g. '3,5,7' ('\='=all) |
| pol | string | None | Select data by polarizations, e.g. 'XX,YY' ('\='=all) |
| dryrun | bool | False | Run the task (False) or display the task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

8.4 hsd_exportdata

The hsd_exportdata task exports the data defined in the pipeline context and exports it to the data products directory, converting and or packing it as necessary.

The current version of the task exports the following products

- a FITS image for each selected science target source image
- a tar file per ASDM containing the final flags version and blparam
- a tar file containing the file web log

TBD

- a file containing the line feature table(frequency,width,spatial distribution)
- a file containing the list of identified transitions from line catalogs

Task Description

Prepare singledish data for export

The hsd_exportdata task exports the data defined in the pipeline context and exports it to the data products directory, converting and or packing it as necessary.

Keyword Arguments

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

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

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

pprfile -- Name of the pipeline processing request to be exported. Defaults to a file matching the template 'PPR_*.xml'.
default: []
example: pprfile=['PPR_GRB021004.xml']

targetimages -- List of science target images to be exported. Defaults to all science target images recorded in the pipeline context.
default: []
example: targetimages=['r_aqr.CM02.spw5.line0.XXYY.sd.im', 'r_aqr.CM02.spw5.XXYY.sd.cont.im']

products_dir -- Name of the data products subdirectory. Defaults to './'
default: "
example: products_dir='./products'

--- pipeline task execution modes

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

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

Output

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

Examples

1. Export the pipeline results for a single sessions to the data products directory
!mkdir ../products hsd_exportdata (products_dir='./products')

Parameter List

Table 58: hsd_exportdata default settings

| Parameter | Type | Default | Description |
|-----------|--------|---------|--|
| pprfile | string | None | The pipeline processing request(PPR) file to be exported |

| | | | |
|----------------------|-------------|-----------|---|
| targetimages | stringArray | None | List of target CASA images to be exported |
| products_dir | string | None | The data products directory |
| pipelinemode | string | automatic | The pipeline operating mode |
| dryrun | bool | False | Run the task (False) or display task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

8.5 hsd_flagdata

The hsd_t_flagdata data performs basic flagging operations on a list of measurements including:

- applying online flags
- apply a flagging template
- autocorrelation data flagging
- shadowed antenna data flagging
- scan based flagging by intent or scan number
- edge channel flagging

Task Description

Do basic flagging of a list of measurement sets

The hsd_flagdata data performs basic flagging operations on a list of measurement sets.

Keyword Arguments

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

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

autocorr -- Flag autocorrelation data.
default: False

shadow -- Flag shadowed antennas.
default: True

scan -- Flag a list of specified scans.
default: True

scannumber -- A string containing a comma delimited list of scans to be flagged.
example: '3,5,6'
default: ''

intents -- A string containing a comma delimited list of intents against which the scans to be flagged are matched.

example: '*BANDPASS*'

default: 'POINTING, FOCUS, ATMOSPHERE, SIDEBAND'

edgespw -- Flag the edge spectral window channels.

default: True

fracspw -- Fraction of the baseline correlator TDM edge channels to be flagged.

default: 0.0625

fracspwfps -- Fraction of the ACS correlator TDM edge channels to be flagged.

default: 0.48387

online -- Apply the online flags.

default: True

fileonline -- File containing the online flags. These are computed by the h_init or hif_importdata data tasks. If the online flags files are undefined a name of the form 'msname_flagonline.txt' is assumed.

default: ''

template -- Apply flagging templates

default: True

filetemplate -- The name of a text file that contains the flagging template for RFI, birdies, telluric lines, etc. If the template flags file is undefined a name of the form 'msname_flagtemplate.txt' is assumed.

default: ''

hm_tbuff -- The heuristic for computing the default time interval padding parameter. The options are 'halfint' and 'manual'. In 'halfint' mode tbuff is set to half the maximum of the median integration time of the science and calibrator target observations.

default: 'halfint'

tbuff -- The time in seconds used to pad flagging command time intervals if hm_tbuff='manual'.

default: 0.0

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

vis -- The list of input measurement sets. Defaults to the list of measurement defined in the pipeline context.

default: ''

flagbackup -- Back up any pre-existing flags.

default: False

--- pipeline task execution modes

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

default: True

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

Output

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

Examples

1. Do basic flagging on a measurement set
hsd_flagdata()

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

Parameter List

Table 59: hsd_flagdata default settings

| Parameter | Type | Default | Description |
|---------------------|-------------|---|---|
| vis | stringArray | None | List of input measurement sets to flag |
| autocorr | bool | False | Flag autocorrelation data |
| shadow | bool | True | Flag shadowed antennas |
| scan | bool | True | Flag specified scans |
| scannumber | string | None | List of scans to be flagged |
| intents | string | POINTING, FOCUS, ATMOSPHERE, SIDE BAND, CHECK | List of intents of scans to be flagged |
| edgespw | bool | True | Flag edge channels |
| fracspw | any | 1.875 GHz | Fraction of baseline correlator edge channels to be flagged |
| fracspwfps | double | 0.048387 | Fraction of ACA correlator edge channels to be flagged |
| online | bool | True | Apply the online flags |
| fileonline | string | None | File of online flags to be applied |
| template | bool | True | Apply a flagging template |
| filetemplate | stringArray | None | File that contains the flagging template |
| hm_tbuff | string | halfint | The time buffer computation heuristic |

| | | | |
|----------------------|--------|-----------|--|
| tbuff | any | 0.0 | List of time buffers (sec) to pad timerange in flag commands |
| qa0 | bool | True | QA0 flags |
| pipelinemode | string | automatic | The pipeline operating mode |
| flagbackup | bool | False | Backup preexistings flags before applying new ones |
| dryrun | bool | False | Run the task (False) or display the command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

8.6 hsd_imaging

Task Description

Generate single dish images

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

Keyword Arguments

mode -- imaging mode controls imaging parameters in the task. Accepts either 'line' (spectral line imaging) or 'ampcal' (image settings for amplitude calibrator)

default: 'line'

options: 'line', 'ampcal',

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

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

default: 'automatic'.

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

infile -- List of data files. These must be a name of Measurementsets that are registered to context via hsd_importdata task.

default: []

example: vis=['uid___A002_X85c183_X36f.ms', 'uid___A002_X85c183_X60b.ms']

field -- Data selection by field names or ids.

default: " (all fields)

example: '*Sgr*,M100'

spw -- Data selection by spw IDs.

default: " (all spws)

example: '3,4' (generate images for spw 3 and 4)

--- pipeline task execution modes

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

default: True

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

default: True

Output

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

Parameter List

Table 60: hsd_imaging default settings

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

8.7 hsd_importdata

Task Description

Imports data into the single dish pipeline

The hsd_importdata task loads the specified visibility data into the pipeline context unpacking and / or converting it as necessary.

Keyword Arguments

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

vis -- List of visibility data files. These may be ASDMs, tar files of ASDMs, MSs, or tar files of MSs, If ASDM files are specified, they will be converted to MS format.

default: []

example: vis=['X227.ms', 'asdms.tar.gz']

session -- List of sessions to which the visibility files belong. Defaults to a single session containing all the visibility files, otherwise a session must be assigned to each vis file.

default: []

example: session=['Session_1', 'Sessions_2']

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

default: 'automatic'.

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

asis -- ASDM tables to convert as is

default: 'Antenna Station Receiver CalAtmosphere'

example: 'Receiver', "

process_caldevice -- Ingest the ASDM caldevice table

default: False

example: True

overwrite -- Overwrite existing MSs on output.

default: False

bdfflags -- Apply BDF flags on line

default: True

lazy -- Use the lazy filter import

default: False

dbservice -- Use online flux catalog on import

default: False

with_pointing_correction -- add (ASDM::Pointing::encoder - ASDM::Pointing::pointingDirection) to the value to be written in MS::Pointing::direction

default: True

--- pipeline task execution modes

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

default: True

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

default: True

Output

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

Examples

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

```
hsd_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e',  

'../rawdata/uid_A002_x30a93d_X44e'])
```
2. Load an MS in the current directory into the context.

```
hsd_importdata (vis=[uid___A002_X30a93d_X43e.ms])
```
3. Load a tarred ASDM in ../rawdata into the context.

```
hsd_importdata (vis=['../rawdata/uid___A002_X30a93d_X43e.tar.gz'])
```
4. Check the hsd_importdata inputs, then import the data

```
myvislist = ['uid___A002_X30a93d_X43e.ms', 'uid_A002_x30a93d_X44e.ms']  

hsd_importdata(vis=myvislist, pipelinemode='getinputs')  

hsd_importdata(vis=myvislist)
```
5. Load an ASDM but check the results before accepting them into the context.

```
results = hsd_importdata (vis=['uid___A002_X30a93d_X43e.ms'],  

acceptresults=False)  

results.accept()
```
6. Run in dryrun mode before running for real

```
results = hsd_importdata (vis=['uid___A002_X30a93d_X43e.ms'], dryrun=True)  

results = hsd_importdata (vis=['uid___A002_X30a93d_X43e.ms'])
```

Parameter List

Table 61: hsd_importdata default settings

| Parameter | Type | Default | Description |
|---------------------------------|-------------|--|--|
| vis | stringArray | None | List of input visibility data |
| session | stringArray | None | List of visibility data sessions |
| pipelinemode | string | automatic | The pipeline operating mode |
| asis | string | Antenna Station Receiver CalAtmosphere CalWVR | ASDM to convert as is |
| process_caldevice | bool | False | Import the caldevice table from the ASDM |
| overwrite | bool | False | Overwrite existing files on import |
| bdf flags | bool | True | Apply BDF flags on import |
| lazy | bool | False | Use the lazy import option |
| dbservice | bool | False | Use the online flux catalog |
| with_pointing_correction | bool | True | Apply pointing correction to DIRECTION |

| | | | |
|----------------------|--------|-----------|---|
| createmms | string | automatic | Create a MMS |
| dryrun | bool | False | Run the task (False) or display task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |

8.8 hsd_k2jycal

Task Description

Derive Kelvin to Jy calibration tables
Derive the Kelvin to Jy calibration for list of measurement sets

Keyword Arguments

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

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

```
MS,Antenna,Spwid,Polarization,Factor
uid__A002_X316307_X6f.ms,CM03,5,XX,10.0
uid__A002_X316307_X6f.ms,CM03,5,YY,12.0
uid__A002_X316307_X6f.ms,PM04,5,XX,2.0
uid__A002_X316307_X6f.ms,PM04,5,YY,5.0
```

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

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

2,2011-11-13 01:45:00,2011-11-13 02:15:00,ANONYMOUS,86243.0,500.0,I,90.0
 2,2011-11-13 01:45:00,2011-11-13 02:15:00,ANONYMOUS,86243.0,1000.0,I,110.0

The line starting with '#' indicates a meta data section and header. The header must exist. The factor to apply is identified by matching the session ID, antenna name, frequency and polarization of data in each line of the file. Note the observation date is supplementary information and not used for the matching so far. The lines whose antenna name is 'ANONYMOUS' are used when there is no measurement for specific antenna in the session. In the above example, if science observation of session 1 contains the antenna PM04, Jy/K factor for ANONYMOUS antenna will be applied since there is no measurement for PM04 in session 1. If no file name is specified or specified file doesn't exist, all Jy/K factors are set to 1.0.

default: 'jyperk.csv'

example: ", 'working/jyperk.csv'

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

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

infile -- List of input visibility files

default: none; example: vis='ngc5921.ms'

caltable -- Name of output gain calibration tables

default: none; example: caltable='ngc5921.gcal'

-- Pipeline task execution modes

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

default: True

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

default: True

Output

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

Example

1. Compute the Kevin to Jy calibration tables for a list of measurement sets

hsd_k2jycal()

Parameter List

Table 62: hsd_k2jycal default settings

| Parameter | Type | Default | Description |
|---------------------|--------|------------|--------------------------------|
| reffile | string | jyperk.csv | File of Jy/K conversion factor |
| pipelinemode | string | automatic | The pipeline operations mode |

| | | | |
|----------------------|-------------|-------|--|
| infile | stringArray | None | List of input measurement sets |
| caltable | stringArray | None | List of output caltable(s) |
| dryrun | bool | False | Run the task (False) or list commands (True) |
| acceptresults | bool | True | Automatically apply results to context |

8.9 hsd_skycal

Task Description

Calibrate data

The hsd_calsky task generates a caltable for sky calibration that stores reference spectra, which is to be subtracted from on-source spectra to filter out non-source contribution.

Keyword Arguments

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

calmode -- Calibration mode. Available options are 'auto' (default), 'ps', 'otf', and 'otfraster'. When 'auto' is set, the task will use preset calibration mode that is determined by inspecting data. 'ps' mode is simple position switching using explicit reference scans. Other two modes, 'otf' and 'otfraster', will generate reference data from scans at the edge of the map. Those modes are intended for OTF observation and the former is defined for generic scanning pattern such as Lissajous, while the later is specific use for raster scan.

default: 'auto'

options: 'auto', 'ps', 'otf', 'otfraster'

fraction -- Subparameter for calmode. Edge marking parameter for 'otf' and 'otfraster' mode. It specifies a number of OFF scans as a fraction of total number of data points.

default: '10%'

options: String style like '20%', or float value less than 1.0.

For 'otfraster' mode, you can also specify 'auto'.

noff -- Subparameter for calmode. Edge marking parameter for 'otfraster' mode. It is used to specify a number of OFF scans near edge directly instead to specify it by fractional number by 'fraction'. If it is set, the value will come before setting by 'fraction'.

default: -1 (use setting by 'fraction')

options: any positive integer value

width -- Subparameter for calmode. Edge marking parameter for 'otf' mode. It specifies pixel width with respect to a median spatial separation between neighboring two data in time. Default will be fine in most cases.

default: 0.5

options: any float value

elongated -- Subparameter for calmode. Edge marking parameter for 'otf' mode. Please set True only if observed area is elongated in one direction.

default: False

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

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

infiles -- List of data files. These must be a name of Scantables that are registered to context via `hsd_importdata` task.
default: []
example: `vis=['X227.PM01.asap', 'X227.PM02.asap']`

field -- Data selection by field name.
default: " (all fields)

spw -- Data selection by spw.
default: " (all spws)
example: '3,4' (generate caltable for spw 3 and 4) ['0','2'] (spw 0 for first data, 2 for second)

scan -- Data selection by scan number.
default: " (all scans)
example: '22,23' (use scan 22 and 23 for calibration) ['22','24'] (scan 22 for first data, 24 for second)

pol -- Data selection by pol.
default: " (all polarizations)
example: '0' (generate caltable for pol 0) ['0~1','0'] (pol 0 and 1 for first data, only 0 for second)

--- **pipeline task execution modes**

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

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

Output

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

Examples

1. Generate caltables for all data managed by context.
`default(hsd_skycal)`
`hsd_skycal()`

Parameter List

Table 63: hsd_skycal default settings

| Parameter | Type | Default | Description |
|----------------------|-------------|-----------|---|
| calmode | string | auto | Calibration mode (default auto) |
| fraction | any | 10% | Fraction of the OFF data to mark |
| noff | int | -1 | Number of the OFF data to mark |
| width | double | 0.5 | Width of the pixel for edge detection |
| elongated | bool | False | Whether observed area is elongated in one direction or not |
| pipelinemode | string | automatic | The pipeline operating mode |
| infile | stringArray | None | List of input files to be calibrated (default all) |
| field | string | None | Field to be calibrated (default all) |
| spw | any | None | Select data by IF IDs (spectral windows), e.g. '3,5,7' ('\ '=all) |
| scan | any | None | Select data by scan numbers, e.g. '21~23' ('\ '=all) |
| dryrun | bool | False | Run the task (False) or display task command (True) |
| acceptresults | bool | True | Add the results into the pipeline context |



The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of the European Organization for Astronomical Research in the Southern Hemisphere (ESO), the U.S. National Science Foundation (NSF) and the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Republic of Chile. ALMA is funded by ESO on behalf of its Member States, by NSF in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and by NINS in cooperation with the Academia Sinica (AS) in Taiwan and the Korea Astronomy and Space Science Institute (KASI).

ALMA construction and operations are led by ESO on behalf of its Member States; by the National Radio Astronomy Observatory (NRAO), managed by Associated Universities, Inc. (AUI), on behalf of North America; and by the National Astronomical Observatory of Japan (NAOJ) on behalf of East Asia. The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.

