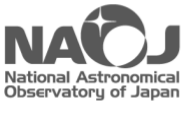
Doc 13.12, version 1.0 | October 2026

ALMA QA2 Data Products for Cycle 13









www.almascience.org

ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), NSTC 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:

Document	Version	Date	Editors
1.12	1.0	Feb 2014	Dirk Petry
2.12	2.0	Sep 2014	Dirk Petry
3.12	1.0	Nov 2015	Dirk Petry
5.12	2.0	May 2018	Dirk Petry, Arielle Moullet, Mark Lacy
7.12	1.0	Feb 2021	Luciano Cerrigone, Dirk Petry
8.2	1.0	Apr 2022	Fabrizia Guglielmetti et al.
9.10	1.0	Sep 2022	Fabrizia Guglielmetti et al.
10.9	1.1	Oct 2023	Fabrizia Guglielmetti et al.
11.3	1.2	Oct 2024	Fabrizia Guglielmetti et al.
12.12	1.0	Oct 2025	Fabrizia Guglielmetti et al.
13.12	1.0	Oct 2026	Fabrizia Guglielmetti et al.

Contributors

Fabrizia Guglielmetti (ESO), Dirk Petry (ESO), Kouichiro Nakanishi (NAOJ), Erica Keller (NRAO), Drew Brisbin (JAO), Luciano Cerrigone (JAO), Hugo Messias (JAO), Jaime Vargas (JAO), Liza Videla (JAO), Harold Francke (JAO), Misato Fukagawa (NAOJ), Andrés Guzmán (JAO), John Hibbard (NRAO), Mark Lacy (NRAO), Brian Mason (NRAO), Arielle Moulet (NRAO), Hiroshi Nagai (NAOJ), Theodoros Nakos (JAO), Tony Remijan (NRAO), Toshiki Saito (NAOJ), Thomas Stanke (ESO), Catarina Ubach (NRAO) and Martin Zwaan (ESO)

In publications, please refer to this document as:

Guglielmetti, F. et al., 2026, ALMA QA2 Data Products for Cycle 13, Doc 13.12, ver 1.0, ALMA

Table of contents

1 Introduction 3

2 Group ObsUnitSet, Member ObsUnitSet, and Scheduling Block 5

3 Overview 5

4 Data Delivery Products 9

4.1 README 9

4.2 Data Reduction Scripts (*script* directory) 10

4.3 Imaging Products (*product* directory) 11

4.4 Calibration Tables (*calibration* directory) 13

4.5 QA Documentation (*qa* directory) 13

4.6 VLBI data products 15

4.6.1 Observing modes and calibration 15

4.6.2 Contents of the VLBI delivery 16

4.6.3 ALMA calibration products for VLBI processing 17

5 Generating the calibrated visibilities or TP data 18

5.1 Running the *scriptForPI.py* 18

5.2 Running the *scriptForPI.py* in the case of manually calibrated data 18

5.3 Running the *scriptForPI.py* in the case of pipeline-calibrated data 23

5.4 Running casa_pipescript.py 26

5.4.1 Considerations for Running Supplemental Manual Imaging Tasks 26

5.5 Saving disk space during and after the execution of the scriptForPI.py 26

5.6 Splitting out the calibrated data at the end of running the scriptForPI.py 27

5.7 Generating the pipeline continuum subtraction products 28

5.8 Generating plots related to the renormalization task 28

5.9 Manually calibrated TP data 29

5.10 Pipeline calibrated TP data 29

6 Relevant documentation and help 31

Acronyms

In the following is a list of acronyms frequently appearing in the text:

ALMA	Atacama Large Millimeter/Submillimeter Array
APP	ALMA Phasing Project
APS	ALMA Phasing System
ARC	ALMA Regional Center
ASDM	ALMA Science Data Model
CASA	Common Astronomy Software Applications package
DP	Data Processing
EB	Execution Block
FOV	Field Of View
GOUS	Group Observing Unit Set
IF	Interferometric
JAO	Joint ALMA Observatory
MOUS	Member Observing Unit Set
MS	Measurement Set
PB	Primary Beam
PI	Principal Investigator
QA	Quality Assurance
QA0	QA level 0
QA1	QA level 1
QA2	QA level 2
SB	Scheduling Block
TP	Total Power
UID	Unique Identification number
VLBI	Very Long Baseline Interferometry

1 Introduction

The objective of ALMA Quality Assurance (QA) is to deliver reliable final data products to the Principal Investigators (PIs) that meet the control parameters outlined in the science goals. This ensures the data is calibrated to the required accuracy and contains no significant calibration or imaging artefacts. As with all previous cycles, imaging products for Cycle 13 will be produced on a "best effort" basis during the QA process.

The first step of QA (QA0) is conducted shortly after observations and is based on metadata and quick-look analysis at the Execution Block (EB) level. Additional procedures are employed to verify data quality in an extended process named QA0+ that involves executing a simplified calibration and imaging pipeline to provide a pseudo-continuum proxy for the final image quality. If no fundamental issues are identified during the initial QA0+ assessment, EBs requiring manual processing are assigned QA0 PASS, while the remaining EBs are processed using the science pipeline. An automated quality assessment of pipeline-processed EBs is performed based on the QA scores. EBs assessed as meeting the required quality criteria are automatically approved through the Fast Lane process and assigned QA0 PASS. EBs that do not meet these criteria are referred to the Slow Lane for manual review, during which a person assesses their quality and determines whether they should be assigned QA0 PASS or classified as QA0 Semipass/Fail. During the first level of QA (QA1), the observatory monitors array and antenna performance parameters to ensure high data quality. The second level of quality assurance (QA2) focuses on data calibration and imaging performed per MOUS: the EBs in a MOUS are calibrated and imaged together and an automatic evaluation of data quality based on QA scores is carried out. Data processing (DP) experts (i.e. Data Reducers and Data Reduction Managers) at JAO, the ARCs and/or the ARC nodes review MOUSs classified as Slow Lane to identify and, where necessary, correct potential issues. DPs experts verify that the observations meet the science goals requested by the PI. In particular, frequency setup, spatial setup, angular resolution, and continuum and/or line detection sensitivity are verified. MOUSs classified as Fast Lane are automatically sent for delivery to the PIs. This document outlines the final science data products that are archived and delivered to the PIs at the conclusion of the QA2 process.

If the requirements are met within the cycle-dependent tolerances (see chapter 11 of the ALMA Technical Handbook), the data are declared "QA2 Pass", packaged in a standardized way, and delivered to the PI.

If the requirements are not met and additional observations of the Scheduling Block (SB) are no longer possible within the cycle time allotted to the project, the data are declared "QA2 Semi-pass" and delivered soon after it becomes clear that re-observation is not possible (i.e., the observing unit has "timed-out"). Data may also be delivered as QA2 Semi-pass if it is determined that the data quality cannot be improved by further observations (for instance, in the case of dynamic-range limited noise[1]), or if all the datasets taken are classified as QA0 Semi-pass and the project has timed out. In the latter case, only the QA0 Semi-pass raw data are delivered.

[1] Please note in case of ALMA Band 1 observations, data may exhibit a bandpass ripple that impacts spectral dynamic range. For more information, see the Knowledge Base article:
https://help.almascience.org/kb/articles/why-do-i-see-a-ripple-in-my-band-1-data

The delivery is made in two parts:

a) *the science products and supporting material,*

b) *the raw data.*

Part *b*, the raw data, consists of visibility or TP datasets in ALMA native format, i.e. the ALMA Science Data Model (ASDM), which are needed if the PI would like to perform custom data calibration. Part *a* of the delivery, the "QA2 Products", consists of the QA2 data reduction scripts, log files, calibration and flagging tables, and the imaging products on which the quality assurance decision was based. This document describes the contents of the Products section of the delivery in detail and is aimed at ALMA users.

Until September 2014, QA2 work was entirely performed manually using the QA2 calibration script generator tool[2]. We refer to data processed in this way as "**manually calibrated"**. In September 2014, the first official version of the ALMA Science Pipeline became available and has since then been upgraded several times. It is now used to calibrate most of the data. We refer to data processed by the Science Pipeline as "**pipeline calibrated**". In addition to the content in the QA2 report, the user can tell whether the data are pipeline calibrated by the presence of a "*PPR*.xml" or a "*pprequest.xml" file in the *script* subdirectory of the delivery package. See Section 4 for more details.

At the beginning of Cycle 4, the Science Pipeline was enabled to also perform imaging. This increased the processing capacity significantly. We refer to data processed by the imaging section of the Science Pipeline as **"pipeline imaged".** If the data are imaged by a data reducer using CASA without the help of the pipeline (typically using templates or the imaging script generator tool[2]), we call them **"manually imaged"**. The approval of the imaging products remains in the hands of the data reducers and Data Reduction Managers. Manual imaging can be necessary even when the pipeline was used for calibration, depending on the specifics of the dataset. Sometimes manual imaging is performed in addition to pipeline imaging, for example to show the PI that the science goal can be reached just by tweaking one of the imaging parameters. In this case, the QA2 report will mention the delivery of **supplemental manual imaging**. In Cycle 10, the pipeline performed both intensity and full polarization calibrations for full polarization data sets, while imaging continued to be carried out manually. Starting in Cycle 12, the pipeline performs full polarization calibration and full Stokes imaging. Another advancement is the implementation of automated self-calibration heuristics for non-ephemeris targets. For more details, see Chapters 10 and 11 of the Cycle 13 ALMA Technical Handbook. Full polarization processing and self-calibration implementations in the pipeline are described at **https://almascience.org/processing/science-pipeline**.

[2] The calibration and imaging script generators are CASA scripts which create a template script for data reduction, which is then edited by the data reducer. See Petry, D. et al. 2014, Proc. SPIE, 9152, 91520J

2 Group ObsUnitSet, Member ObsUnitSet, and Scheduling Block

In ALMA's data structure, the smallest scheduling entity processed and delivered individually is the Member Observing Unit Set (MOUS). Each MOUS is the collection of all executions of one particular SB. The QA2 is therefore conducted on an individual MOUS basis for all pass Execution Blocks (EBs) associated with a single SB. Products for each MOUS are delivered with a unique label containing the MOUS UID, i.e. a hexadecimal identifier that provides access to the ALMA archive.
MOUSs are organized into Group Observation Unit Sets (GOUSs). A GOUS is the set of all MOUSs which are meant to be combined to obtain the final image products of a given science goal. The GOUS corresponds exactly to one science goal. In Cycle 13, the following types of MOUSs may comprise a single GOUS:

a) Up to two 12-m Array MOUS(s),

b) Up to one 7-m Array MOUS,

c) Up to one TP observation MOUS.

The delivery packaging for each MOUS results in three tar files named as follows:

<Project ID>_<MOUS_UID>_001_of_001.tar
<Project ID>_<MOUS_UID>_auxiliary.tar
member_<MOUS_UID>_README.txt.tar

The archive offers the possibility to download individual product files. The auxiliary tar file contains calibration tables, logs, scripts and quality assurance reports, including the weblog, which is the complete record of the pipeline processing of one MOUS in the form of a system of web pages. The tar file marked with 001_of_001 contains the FITS product files of the targets and possibly the calibrators. The README.txt.tar file contains a description of the main actions, categories and files of the ALMA data at the MOUS level. Before Cycle 4, the README file was providing information on what is delivered and how to work with it. This information has been transferred to the QA2 Report since Cycle 5.
Optionally, if the user would like to recreate the calibrated data, the user will need to download also the uncalibrated raw data and these will appear as tar files named

<Project ID>_<EB_UID>_asdm.sdm.tar

where EB_UID indicates the unique ID that identifies the execution block, since one tar file will be available per SB execution. By untarring any of the tar files in a delivery package, the user will obtain a directory tree as described in the next Section.

3 Overview

An ALMA data delivery in Cycle 13 consists of the six items shown in the directory tree below. Items 1 to 5 are the main delivery package (*part a*), which should be downloaded first. Item 6 (*part b*) is only required if the users would like to work with the raw visibility or TP data. Once unpacked, all data fall into a standardized directory structure:

```
|-- project_id/
|   |-- science_goal.ouss_id/
|   |   |-- group.ouss_id/
|   |   |   |-- member.ouss_id/
|   |   |   |   |-- product/        (1)
|   |   |   |   |-- calibration/    (2)
|   |   |   |   |-- qa/             (3)
|   |   |   |   |-- script/         (4)
|   |   |   |   |-- log/            (5)
|   |   |   |   |-- raw/            (6)   (only present when part b is unpacked)
```

The contents of the directory are summarized below. When an item is marked as "best effort", it means that it is generated by the data reducers only if sufficient resources are available and it is compatible with the science goal.

1) Imaging Products (in FITS format) in *product* subdirectory

a) for pipeline imaged data:

A set of continuum and line images for all science spectral windows, covering as large as possible a part of the spectral range. This set is as complete as possible given the computing resources of the processing pipeline, i.e. for datasets with very high spatial and/or spectral resolution some parts of the set may be automatically omitted by the pipeline, to keep the size of the product package within reasonable limits. This process is referred to as the "mitigation" of data products. From Cycle 10, self-calibrated imaging products are characterized by **selfcal** extension, to be distinguished by the regular imaging products. The regular imaging products generated by the pipeline include the **regcal** extension. However, during the ingestion process into the science Archive, the **regcal** extension is omitted from the regular imaging product filenames to maintain consistency with filenames from previous cycles, when self-calibration had not yet been included as part of the pipeline. In Cycle 11, cube imaging products are delivered with self-calibration applied whenever possible; if self-calibration cannot be applied, only the regcal products are provided. Starting in Cycle 12, full Stokes continuum and line cube images are generated by the pipeline for data observed in the full polarization mode. The pipeline generates Stokes I science images to obtain a mask to use for full Stokes imaging stages. Stokes I science target images are not delivered for full polarization data. Self-calibration is not attempted for full Stokes imaging. Full Stokes are indicated in the image name by the term **IQUV.** For comprehensive information regarding the pipeline output, including detailed descriptions of its processes and functionalities, please refer to the official ALMA pipeline documentation.

This resource can be accessed through: **https://almascience.org/processing/science-pipeline**

b) for pipeline calibrated data:

Continuum images of the bandpass and phase calibrators and (when present) the check source(s) and polarization calibrator, for each of the science spectral windows for interferometric data (12-m

and 7-m Arrays) only. Full Stokes continuum images are delivered for the polarization calibrator when the data are observed in full polarization mode.

c) manually imaged line cubes:

For the representative spectral window: One spectral cube of "representative channels" for at least the representative target in the MOUS, made at the requested spectral resolution considering the bandwidth specified for sensitivity. If the achieved spatial resolution is higher than requested, tapering may be applied if needed, to reach the science goal. Coarse continuum subtraction is applied for sources with bright continuum.

— (best effort) For each source and each spectral window: cubes of all species specifically listed in the proposal, made at an angular resolution as close as possible to the requested one and including line-free channels on either side. Continuum subtraction is applied for sources with bright continuum. A note may be added if additional species are found in the data, but imaging may not be carried out.

d) manually imaged continuum and or aggregate bandwidth images:

— One image of all continuum spectral windows, all non-edge channels, for at least the representative target in the MOUS.
— (best effort) One image of all continuum spectral windows, all non-line channels.

2) CASA Tables (in CASA Table format) in *calibration* subdirectory

a) for pipeline calibrated data

- Calibration tables: including Tsys, Antpos, WVR, Gain, Bandpass, Amplitude, Polarization (*caltables.tgz)
- List of the applycal commands for the delivered calibration table (*calapply.txt)
- Flagversions tables (*flagversions.tgz)
- Auxilliary products (*auxproducts.tgz):
  - Calibrator fluxes from the ALMA calibration database (flux.csv)
  - Calibration table antenna positions (antennapos.csv or antennapos.json)
  - Continuum frequency ranges (cont.dat)
  - Line and continuum frequency ranges identified during hif_findroi in *findroi_products.tar
  - Flagging commands added manually by the pipeline operator (*flagtemplate.txt, *flagtargetstemplate.txt, *flagtsystemplate.txt)
  - Pipeline *.selfcal.json restore file for self calibration
  - Pipeline *.timetracker.json and pipeline_stats*.json containing statistical information on the pipeline execution
  - "Kelvin to Jansky" calibration factors for TP Array observations (jyperk_query.csv or jyperk.csv).

See **https://almascience.org/processing/science-pipeline** for more information on the role of these files.

b) for manually calibrated data

— Calibration tables: including Tsys, Antpos, WVR, Gain, Bandpass, Amplitude, Polarization (if applicable)

— Flagversions tables

— Calibration plots and wvrgcal diagnostic output.

3) QA documentation in *qa* subdirectory

a) for pipeline-calibrated and/or imaged data

— The Pipeline Weblog, i.e. a system of webpages containing all the diagnostic plots and other information generated by the pipeline. This is the main source of information on all properties of the data and the steps taken to calibrate/image and flag them. The weblog also contains the summary of the commands executed by the pipeline, the casa log file for each stage and its QA scoring.

— A QA0 Report for each EB UID in pdf format. This document is a summary of what occurred during data acquisition. It contains information about the system and receiver temperature in every antenna, the amount of time spent on calibrators and target sources, and a first quality evaluation of the data.

— A QA2 Report for the MOUS UID in pdf format. This report contains information about the quality assessment performed after data processing, as well as comments from the reviewer responsible for approving the dataset for delivery. Starting from Cycle 10, the observatory has implemented automatic delivery of MOUSes that meet quality assurance standards without any detected issues. In such cases, the reviewer comments are autogenerated, confirming that no problems were encountered during the calibration or imaging processes. The report also contains Instructions on the contents of the data package and guidelines for restoring the data.

b) for manually calibrated/imaged data

— QA Reports for each EB of the imaged data (png and txt files).

— The Manual Weblog is provided in html format. The Manual Weblog contains the information of the CASA reduction scripts and related images: CASA reduction scripts and diagnostic plots are displayed, fits images are listed. Scripts can be downloaded.

— If pipeline tasks were used in the manual imaging script, there may also be the "imaging_weblog.tgz" file. Unpack and open with a browser like the normal Pipeline Weblog mentioned under (a) above.

4) Data Reduction Scripts (ASCII files) in *script* subdirectory

a) for pipeline calibrated and imaged data

— The Python scripts needed to restore the calibrated MeasurementSet(s) (MS(s)) (*scriptForPI.py and *casa_piperestorescript.py) or rerun the entire pipeline (*casa_pipescript.py).

— The pipeline processing request file (PPR*.xml or *pprequest.xml).

b) for pipeline calibrated and manually imaged data

— (in addition to **a)**): The imaging preparation script (if necessary) and the imaging script.

— Optionally also the *scriptForPolCalibration.py

c) for manually calibrated and imaged data

— The Python scripts needed to restore the calibrated MS(s) (*scriptForPI.py, *scriptForCalibration.py, *scriptForSDCalibration.py and optionally *scriptForImagingPrep.py, *scriptForPolCalibration.py)

— The Imaging script (*scriptForImaging.py or *scriptForSDImaging.py).

5) Calibration and imaging log files in *log* subdirectory

— CASA log files from the QA2 processing in case of manually calibrated or manually imaged datasets.

— CASA commands log from the QA2 processing in case of pipeline reduced datasets.

6) Datasets (in ASDM format) in *raw* subdirectory (if downloaded)

— (for optional download) The EBs that were used in the calibration and imaging and the subsequent QA2 assessment.

— (for optional download if applicable) Additional datasets that were NOT used in the data reduction but may still contain valuable scientific information.

4 Data Delivery Products

4.1 README

The complete name of this file in normal deliveries is member.<MOUS_UID>.README.txt. Until the end of Cycle 4, this was a text file summarizing the QA2 results and explaining the structure of the tarball and file naming conventions. Since Cycle 5, this README file has normally served merely as a reference indicating where one can find additional information. A summary of the QA2 results, an explanation of the tarball structure, and the file-naming convention are provided in the QA2 report, located in the *qa* directory.

4.2 Data Reduction Scripts (*script* directory)

In the *script* subdirectory, the user finds the CASA data reduction scripts (Python) that were used to calibrate and image the data. Depending on whether the Science Pipeline was used or the data were calibrated manually, the contents of this directory vary:

1) *uid....ms.scriptForCalibration.py*: one calibration script for each EB of manually calibrated interferometric data.
2) *uid....ms.scriptForSDCalibration.py*: one calibration script for each EB of manually calibrated single-dish data.
3) *scriptForImagingPrep.py* (for manually imaged datasets, if needed): this script contains all necessary steps to prepare imaging that must be performed on all calibrated EBs together (and therefore cannot be performed by the calibration script for the individual EBs). These steps may include alignment of frequency grids before concatenation, concatenation into one MS, adjustments to the flux calibration, etc. The result is an MS or a set of MSs that are ready for final imaging. If present, this script is run automatically as the last step by the *scriptForPI.py* (see below).
4) *scriptForPolCalibration.py* (for manually calibrated full-polarization data): a special calibration routine for full-polarization data executed after the manual per-EB-calibration scripts or pipeline total intensity calibration. If present, it will be run automatically by *scriptForPI.py* (see below).
5) *scriptForImaging.py* (for interferometric data): this script generates the same imaging products (see Section 5) as those stored in the *product* subdirectory of the delivery. Where convenient, the *scriptForImaging.py* may also include steps that could otherwise appear in *scriptForImagingPrep.py*; in such cases, a separate *scriptForImagingPrep.py* is not created.
6) *scriptForSDImaging.py* (for manually imaged single-dish data): this script generates the same imaging products (see Section 5) as those that are stored in the *product* subdirectory of the delivery.
7) ***scriptForPI.py***: in all cases, whether pipeline or manually calibrated, run this script in the *script* directory to regenerate the calibrated MS(s) of the delivered data after downloading the raw ASDMs. The README file and the QA2 report contain instructions (or references) on how to perform this procedure, which is also described in this document. Running *scriptForPI.py* requires that you have downloaded the ASDM datasets and unpacked them so that they reside in the *raw* directory (created during unpacking). See next Section for more details.

   Until mid-Cycle 5, the only type of dataset that was *not* delivered with a *scriptForPI.py* was the *manually calibrated TP* data. Since mid-Cycle 5, all datasets are delivered with scriptForPI.py (see Section 5).
8) *casa_piperestorescript.py*: this script is used by *scriptForPI.py* in case of pipeline-calibrated data to restore the calibrated MS(s) from the raw ASDMs by applying the necessary calibration tables. If this script is not provided or removed, *scriptForPI.py* will run the *casa_pipescript.py* instead.

9) *casa_pipescript.py*: performs the complete calibration from scratch using Pipeline tasks. See the ALMA Science Pipeline User's Guide documentation for more details.

10) The pipeline processing request (PPR): an XML file used to drive the Pipeline calibration and/or imaging. See **https://almascience.org/processing/science-pipeline**

A script named *scriptForFluxCalibration.py* may be present in the *script* directory; if so it will be executed by *scriptForPI.py* before *scriptForImagingPrep.py*. During Cycle 7, this script was employed with CASA 6.1 to evaluate the requirement of rescaling due to strong astronomical lines detected and to apply a flux offset correction when needed. This renormalization process is evaluated by the pipeline from CASA 6.2 onwards (see Section 5.7). Furthermore, the following script file may be present for solar observations: *sun_reduction_util.py* (contains helper routines specific to solar data analysis). In addition to the scripts, *manifest.xml and *pldriver_report.xml are products of the pipeline execution, containing information about the packaged files and metadata, as well as the list of associated EBs.

4.3 Imaging Products (*product* directory)

The *product* subdirectory of the delivery contains the imaging products that were generated either by the imaging section of the Science Pipeline or by a data reducer with the script scriptForImaging.py. These imaging products were created from the calibrated dataset produced when the scriptForPI.py is run. For interferometric data, this directory also contains calibrator images.

A coherent **file naming scheme** is used for all imaging products:

<MOUS>.<target>_<target type>.spw<SPWs>.<imagetype>.<y>.<stokes>.<cmt>.<x>.<ext>

Where

<MOUS>	Is the MOUS UID (for example, uid___A001_X23a_X2c).
<target>	Is the name of the imaged field (it can be the name of a calibrator or of the scientific target).
<target type>	Is "sci" for science target, "ph" for phase calibrator, "bp" for bandpass calibrator, "amp" for flux calibrator (this will not be present if the bandpass calibrator is also used as flux calibrator), "chk" for check source, "pol_leak" for the polarization calibrator.
<SPWs>	Are the imaged spectral window IDs.
<image type>	Is "mfs" for target per-SPW continuum image, "cube" for a spectrally resolved image, "repBW" for cube at representative bandwidth, "cont" for an image with the line channels excluded (in pipeline calibrated/imaged data, this will have all spectral windows merged). For multi-term wideband imaging, additional .tt0,.tt1, etc suffixes indicate Taylor terms, plus .alpha for spectral index image, .alpha.error for error on spectral index estimate (error map), .beta for spectral curvature if nterms > 2.

<y>	From Cycle 10, the image type as outcome from regular or self calibration (y==’selfcal’ for self-calibrated images, while y is blank for regular calibration images).
<stokes>	The name of the image Stokes parameter(s) (e.g., “I”, “IQUV”; or “P” and “A” for polarization intensity and angle).
<cmt>	Can be an explanatory text, such as a molecular transition.
<x>	Is “pbcor” for the primary beam corrected image, “pb” for the PB itself, “mask” for the mask image, “sd” for single dish.
<ext>	Is the filename extension describing the file type, e.g. “fits”.

Note that the ***<cmt>*** field typically contains the word “manual” if the image was obtained by manual imaging. Information about the ALMA naming scheme can also be found in the following knowledgebase article
https://help.almascience.org/kb/articles/what-calibration-and-imaging-products-will-be-delivered-to-me
The goal of ALMA QA2 in Cycle 13 remains that the calibration is reliable and "science ready", but the imaging products may not be. A sufficient fraction of the possible imaging products are created to verify that the data meet the science goals set by the PI (resolution and sensitivity), within some tolerance, but the **imaging may be incomplete**. In general, investigators should expect to need to re-image their data using the provided scripts (for manual imaging) or the pipeline documentation (for pipeline imaged data) as a guideline. In particular, care should be taken to optimize continuum subtraction (if relevant) and fine-tune the image deconvolution parameters (e.g. clean masks and thresholds). See “ALMA Imaging Pipeline Reprocessing” in CASA guides:
https://casaguides.nrao.edu/index.php?title=ALMA_Imaging_Pipeline_Reprocessing

Images conform to the FITS standard 3.0. The imaging clean masks are provided as gzipped FITS files. These represent the final mask used in the last cleaning iteration. The masks used in previous iterations may have been different (smaller in area).
The science images included in deliveries of interferometric data are corrected for the primary beam (PB), i.e. the dependence of the instrument sensitivity on direction within the field of view (FOV). For each image, two files are delivered:

a) The PB-corrected image (file name ending in ".pbcor.fits").

b) The image of the PB used for the correction ("*.pb.fits", possibly gzipped).

The image noise is measured in the uncorrected image. The corrected image (a) is then obtained by dividing the uncorrected image by the PB image (b). The uncorrected image can be recovered using the CASA task “impbcor” in mode "m":

```
impbcor(imagename='image.pbcor.fits',pbimage='image.flux.fits', mode='m',
outfile='image.recovered')
```

For **Full Polarization** data, full Stokes continuum and cubes are delivered. Linear polarization intensity and polarization angle images are no longer part of the delivery package from Cycle 13 and on. If the users would like to have these images, they can be made from the full Stokes images using the CASA task "imstat". These images can also be made using the Cube Analysis and Rendering Tool for Astronomy (CARTA). See **https://cartavis.org/** for more details.
Linear polarization (pol *I*) is calculated as:

$$polI = \sqrt{(Q^2 + U^2)}$$

Stokes V is ignored. No debiasing is applied by default.
Polarization angle (pol *A*) is calculated as:

$$polA = \frac{1}{2}\tan^{-1}\frac{U}{Q}$$

A threshold (polithresh) may be applied to mask linear polarization values below the set threshold.

For **TP** data, images are delivered accounting for the convolution of the sky intensity distribution with the PB of the antenna, implying no need for a PB correction. In early cycles and high frequency regular observations, an additional amplitude calibrator observation was taken whose images were not included.

If the data reducer deemed it necessary, in the case of pipeline imaged data, **supplemental manual imaging** may be added to the delivery package. The user will then find in the *product* directory a file named *manual_imaging.tgz* containing the CASA log files and the scriptForImaging.py used to perform such supplemental imaging. The supplemental products will contain the comment *manual* in their names and can be found either inside the *manual_imaging.tgz* or simply in the *products* directory along with the pipeline products.

4.4 Calibration Tables (*calibration* directory)

All calibration tables generated by the calibration process are stored in the *calibration* subdirectory of the delivery.
In the case of manually calibrated data, information on how each table was created can be found in the scriptForCalibration.py and the scriptForImagingPrep.py scripts. CASA provides functionality to view these tables. Please refer to the CASA documentation (see section 6, in particular the ALMA Science Pipeline User's Guide) on how to do that. However, many of the plots of interest in this context are also available in the QA documentation (*qa* directory, Section 4.5).

In the case of pipeline-calibrated data, the files in this directory are needed to restore the calibrated MS with "*casa_piperestorescript.py*" which is called by "*scriptForPI.py*". For details on how they were generated, please refer to **https://almascience.org/processing/science-pipeline**. All diagnostic plots are available in the Pipeline Weblog, which can be found in the *qa* directory (see Section 4.5).

4.5 QA Documentation (*qa* directory)

In the *qa* subdirectory of the delivery, the user finds diagnostic information obtained during calibration and imaging.

In the case of *pipeline calibrated data,* all diagnostic information is assembled in a system of html pages which is called the **Weblog**. To access the Weblog, run

tar xzf *hifa*weblog.tgz

in the *qa* directory and then navigate to the root of the untarred weblog directory e.g. pipeline-20220914T170517/html and **run h_weblog()** within a CASA session containing pipeline tasks. You can also open the weblog without CASA: from the html directory, run

python3 -m http.server 8080 --bind localhost.

You can then open the weblog in your web browser. If you have multiple weblog sessions open simultaneously, use a different port number for each session, for example 8081 instead of 8080. It is recommended to set your default browser to Firefox to view the weblog, although since the 2021 pipeline release there has been limited support for Chrome and Safari; for more information refer to **https://help.almascience.org/kb/articles/what-is-the-best-way-to-view-the-weblog**.

For more information on the contents of the Weblog, please refer to the Pipeline documentation at **https://almascience.org/processing/science-pipeline**.

In the case of *manually calibrated* data, there are several png images and a textfile for each EB, labelled with the EB UID. The png files can be viewed with standard system tools such as “eog” (under Linux) or with the command “open” under MacOS. They show a set of diagnostic plots describing the following aspects of the data:

1) Observing Schedule (observation intent vs. time).
2) Mosaic Pointing Configuration.
3) Antenna Array Configuration.
4) Effectiveness of the WVR correction for each antenna (for 12M observations).
5) Temporal gain calibration solutions for each antenna.
6) Temporal phase calibration solutions for each antenna.
7) Average bandpass solution for each spectral window.
8) System Temperature vs. frequency for each antenna and each spectral window.
9) Bandpass calibrator amplitude and phase vs. frequency for each spectral window and polarization.
10) Flux calibration model and data (visibility amplitude vs. *uv*-distance).
11) Target visibility amplitude vs. *uv*-distance for each spectral window and polarization.
12) Phase calibrator amplitude and phase vs. frequency for each spectral window and polarization.
13) Target field *uv*-coverage.
14) Test image of the target.
15) Target imaging synthesized beam (PSF).
16) Phase stability plot (PhaseRMS vs. baseline length).

The content of the uid*textfile.txt file is mostly self-explanatory. Of particular interest is the “Check of a target image and sensitivity”, which mentions approximately the achieved spatial resolution and sensitivity. Note that the finally achieved resolution and sensitivity could be different, due to additional procedures in imaging.

From Cycle 7, all available png images and scripts related to the manual data reduction are assembled in the manual Weblog, a system of html pages. To access the Weblog, run

```
tar xzf *manual*weblog.tgz
```

in the *qa* directory and then use your favourite web browser to open the resulting file:

```
manual*/html/index.html
```

In case of manual calibration, all EB references in this weblog are expandable links with the related diagnostic information. In case of manual imaging, the imaging tab provides the related imaging scripts and list of plots. Each script name is a link to visualize the content.

Based on the calibration tables in the *calibration* directory, the user can produce additional diagnostic plots in CASA using the tasks *plotms* and *plotbandpass*. See the CASA documentation for more details.

If additional diagnostic plots on the visibility data themselves are required, one needs first to generate the calibrated MS from the raw data (see Section 5). Once this is done, the CASA task *plotms* can be used to generate the plots.

4.6 VLBI data products

For VLBI observations, ALMA operates as a phased array within a network of telescopes. QA2 processing uses the interferometric data recorded by the ALMA correlator to calibrate ALMA as a VLBI station. The delivery contains calibration material, data reduction scripts, diagnostic information, and images made from the ALMA interferometric data.

The ALMA QA2 products described here should be distinguished from the visibilities formed by correlating the recorded VLBI signals between ALMA and the other telescopes.

As described in Chapter 11 of the **Cycle 13 ALMA Technical Handbook**, VLBI QA2 focuses on obtaining the best possible calibration from the observations that were taken. VLBI observations are not re-observed as a consequence of deficiencies found during QA. The polarization calibration is also designed to support the conversion of the VLBI-correlated data from ALMA's linear polarization basis to the circular basis used for subsequent VLBI analysis. The QA2 report should be consulted for the QA2 result and limitations of the individual delivery.

4.6.1 Observing modes and calibration

The ALMA data contain two subsets of scans: ALMA-mode scans, for which the phasing system is OFF, and APS-mode scans, for which the phasing system is ON. APP refers to the ALMA Phasing Project (this nomenclature is used in the scan intents); APS refers to the ALMA Phasing System and is also used as a label in some delivered filenames.

The polarization response differs between these modes, and their polarization calibration must be treated separately. The scans may be combined after the appropriate full polarization calibration has been applied to each subset. Poor parallactic-angle coverage of the polarization calibrator in ALMA mode can produce less reliable solutions than those obtained from APS-mode scans. Where this limitation is identified in the delivery documentation, users should use the APS-mode scans for

polarization analysis. The delivered calibration scripts specify the scan selections and calibration strategy used during QA2.
The phased-array flux-density calibration uses a primary flux calibrator and self-calibration. Conventional antenna-based Tsys corrections are not applied directly, so that the gain calibration remains consistent with the unweighted sum of antenna signals used for VLBI. Valid Tsys measurements are nevertheless used to estimate residual opacity corrections after self-calibration.

These corrections are source dependent and constant over the observing epoch, and are applied through a table such as ***flux_inf.APP.OpCorr*** (Goddi et al. 2019, Appendix A1). Except for earlier cycles, whenever the target is actively phased, the phase calibrator observed in ALMA-mode scans is the science target itself. This has been adopted so that the PIs can have an alternative way to compute the source flux.

4.6.2 Contents of the VLBI delivery

Calibration material and diagnostic plots are packaged by SB in ***<SB_name>.calibration.tgz*** and ***<SB_name>.calibration.plots.tgz***. Antenna plots with the suffixes ***.plotants.ALL.png*** and ***.plotants.APP-only.png*** distinguish the full antenna set from the phased subset.

The VLBI image set contain per-spectral-window Stokes I, Q, U, and V images made from APS-mode scans, for science targets and calibrators. The APS label in filenames identifies this phased-array scan selection. Deconvolution masks are packaged as ***.mask.tgz*** archives. The actual source and scan selections are recorded in *scriptForImaging.py*; the general image conventions are described in Section 4.3 (above). For example, a full Stokes continuum image from APS-mode scans are named:

<MOUS>.<source>_<type>.spw<N>.mfs.IQUV.APS.pbcor.fits

For VLBI products, mfs identifies multi-frequency synthesis imaging, IQUV identifies the Stokes parameters, and APS labels the phased-array scan subset. The ending ***.pbcor.fits*** identifies a primary-beam-corrected image. A corresponding ***.pb.fits*** file contains the primary beam used for the correction. Deconvolution masks are packaged as ***.mask.tgz*** archives in VLBI deliveries. Their format should be checked after unpacking.

4.6.2.1 Polarization calibration

There are two subsets of observations in these ALMA data: the “ALMA-mode” scans (observed with the phasing system OFF) and the “APS-mode” scans (observed with the phasing system ON). The two modes are not compatible, in terms of the polarization calibration. However, both modes can be combined after the full-polarization calibration. Sometimes, a poor parallactic-angle coverage of the polarization calibrator in the ALMA-mode scans may degrade the ALMA-mode calibration and make these data (much) worse than those of the APS-mode scans. In those cases, for polarization calibration please use only the APP-mode scans (which typically have a better parallactic-angle coverage of the polarization calibrator).

4.6.2.2 Flux scale error

The absolute flux-density scale has been derived from self-calibration based on a primary flux-density calibrator (e.g., source 3C279). The system temperatures (Tsys) are not used in the phased-array data

calibration. The reason for this choice is that the phased signal for VLBI is an unweighted sum of the signals from all the phased antennas (with no Tsys applied), therefore all phased antennas should be treated equally in the gain calibration for PolConvert. The bulk of the opacity effects is removed with self-calibration, whereas Tsys would correct for second-order opacity effects, related to the difference between the opacity correction in the observation of the primary flux calibrator and the (average) opacity in the observation of any given source. These second-order opacity effects are accounted for by applying a global scaling factor, which is source-dependent and **constant** during the observing epoch, estimated using the antenna-wise average of valid Tsys measurements, as described in Appendix A1 of Goddi et al. 2019, PASP, DOI: 10.1088/1538-3873/ab136a. This was implemented by applying the calibration table ***flux_inf.APP.OpCorr** (the one also used in **uid*scriptForCalibration.py** scripts).

The amplitude calibration provided in the package is the same as that of the ALMA-VLBI data, which is provided as a set of ANTAB (AIPS) tables generated by the program PolConvert. This program is run by the VLBI correlators and, besides generating the calibration information for ALMA, it transforms the linear-polarization ALMA-VLBI data streams into a circular basis for VLBI.
A complete description of the phase and amplitude calibration of ALMA data in VLBI mode can be found in: Goddi et al. 2019, PASP, DOI: 10.1088/1538-3873/ab136a
A full description of PolConvert can be found in: Marti-Vidal et al. 2016, A&A, 587, 143

4.6.2.3 Imaging of the calibrators

The *products* directory contains FITS images of all the sources observed in the EBs of the project. In the delivered products, only the images from the APS-mode scans are included.

4.6.3 ALMA calibration products for VLBI processing

The ALMA calibration information is also used by PolConvert in the VLBI correlator workflow. PolConvert applies the ALMA station calibration, converts the mixed-polarization VLBI visibilities to a circular basis, and generates ALMA amplitude-calibration information in ANTAB format (Martí-Vidal et al. 2016; Goddi et al. 2019).

The ASDMs and images described here contain ALMA interferometric data. Reconstructing these MSs does not reconstruct baselines to other VLBI stations. Bad-quality ALMA data can also affect VLBI baselines to ALMA at the corresponding times and frequencies, so the data reducer comments should be considered alongside the VLBI quality assessments.

Note that VLBI schedules (i.e., tracks) regularly include more than one ALMA schedule block, sometimes from a different project. Whenever the resulting per-MOUS calibration fails, the data reducer will attempt to calibrate the track as a whole. In such a case, the calibration tables apply to more than one MOUS and individual PIs will not be able to regenerate them only from the ASDMs associated to their project. As a result of this workflow, the scripts in the deliverables do not regenerate calibration tables, but rather apply them. Another potential effect is that the product images for specific targets will be deeper than the ones one can get from per-MOUS ASDMs. Finally, we note that, in some cycles, there was the need to interleave projects. This results in science targets

being shared between projects and/or MOUS. However, the image products will only contain the image corresponding to the MOUS-specific science target.

5 Generating the calibrated visibilities or TP data

In ALMA Cycle 0 and early Cycle 1, the visibility data were delivered with all other products in ready-to-use MS format. Since the middle of Cycle 1, the products are separated from the raw visibility or TP data (as described in the introduction); downloads of the latter are now optional. To minimize download time, the raw data are delivered in the native ALMA format, the ASDM.

If the user would like to only modify or extend the imaging of the dataset, but accepts the observatory calibration of the data, they need to download the raw data and apply the calibration via the scripts provided with the delivery as described in the following. Otherwise, the user can download the raw data and use the calibration scripts or pipeline documentation as a guidance to develop a custom calibration. Moreover, each ARC provides calibrated Measurement Sets directly:
https://help.almascience.org/kb/articles/how-do-i-obtain-a-file-of-calibrated-visibilities-measurement-set-for-alma-data

The download of the raw data depends on the details of where the data were staged for delivery. The user should refer to the information in the delivery email as to how exactly the raw data should be obtained. Unless there is a technical problem, this will work via the ALMA Archive Request Handler: See **http://almascience.org/alma-data**.

5.1 Running the *scriptForPI.py*

Both for manually and pipeline-calibrated data, the calibrated visibilities are restored by unpacking the tarballs obtained from the Archive into a designated location, navigating to the "script" directory, launching the appropriate CASA version, and running <member...scriptForPI.py> via the execfile command. This process is described in detail in the following sections.

5.2 Running the *scriptForPI.py* in the case of manually calibrated data

For manually calibrated data, the scriptForPI.py will find the scripts named scriptForCalibration.py and, e.g., scriptForImagingPrep.py (if present), set up every necessary item for their execution, and then run them to obtain the MS ready for imaging.
Once the data-products tar file is downloaded and unpacked, the user finds a certain directory structure on disk (see the description in Section 3). The following directory tree shows an example of manually calibrated (rather than pipeline calibrated) data. However, also for pipeline calibrated data, the procedure for running "*scriptForPI.py*" is the same and has a similar final product, although the naming differs.

```
2026.1.01573.S
└── science_goal.uid___A001_X3621_X53f
   └── group.uid___A001_X3621_X540
      └── member.uid___A001_X3621_X541
```

```
├── calibration
│   ├── uid___A002_X110fd13_X12b88.calibration.plots.tgz
│   ├── uid___A002_X110fd13_X12b88.calibration.tgz
│   ├── uid___A002_X110fd13_X12b88.ms.wvrgcal.txt
│   ├── uid___A002_X110fd13_X131c8.calibration.plots.tgz
│   ├── uid___A002_X110fd13_X131c8.calibration.tgz
│   ├── uid___A002_X110fd13_X131c8.ms.wvrgcal.txt
│   ├── uid___A002_X110fd13_X1b52b.calibration.plots.tgz
│   ├── uid___A002_X110fd13_X1b52b.calibration.tgz
│   ├── uid___A002_X110fd13_X1b52b.ms.wvrgcal.txt
│   ├── uid___A002_X110fd13_X230eb.calibration.plots.tgz
│   ├── uid___A002_X110fd13_X230eb.calibration.tgz
│   ├──uid___A002_X110fd13_X230eb.ms.wvrgcal.txt
│   ├── uid___A002_X110fd13_X59fd.calibration.plots.tgz
│   ├── uid___A002_X110fd13_X59fd.calibration.tgz
│   └── uid___A002_X110fd13_X59fd.ms.wvrgcal.txt
├── log
│   ├── member.uid___A001_X3621_X541*.log.tgz
│   ├── uid___A002_X110fd13_X12b88.log.tgz
│   ├── uid___A002_X110fd13_X131c8.log.tgz
│   ├── uid___A002_X110fd13_X1b52b.log.tgz
│   ├── uid___A002_X110fd13_X230eb.log.tgz
│   └── uid___A002_X110fd13_X59fd.log.tgz
├── member.uid___A001_X3621_X541.README.txt
├── product
│   ├── member..J2206-0031_sci.spw25.cube.I.mask.fits.gz
│   ├── member..J2206-0031_sci.spw25.cube.I.pb.fits.gz
│   ├── member..J2206-0031_sci.spw25.cube.I.pbcor.fits
│   ├── member..J2206-0031_sci.spw25.mfs.I.mask.fits.gz
│   ├── member..J2206-0031_sci.spw25.mfs.I.pb.fits.gz
│   ├── member..J2206-0031_sci.spw25.mfs.I.pbcor.fits
│   ├── member..J2206-0031_sci.spw25_27_29_31_33.cont.I.mask.fits.gz
│   ├── member..J2206-0031_sci.spw25_27_29_31_33.cont.I.pb.fits.gz
│   ├── member..J2206-0031_sci.spw25_27_29_31_33.cont.I.pbcor.fits
│   ├── member..J2206-0031_sci.spw27.cube.I.mask.fits.gz
│   ├── member..J2206-0031_sci.spw27.cube.I.pb.fits.gz
│   ├── member..J2206-0031_sci.spw27.cube.I.pbcor.fits
│   ├── member..J2206-0031_sci.spw27.mfs.I.mask.fits.gz
│   ├── member..J2206-0031_sci.spw27.mfs.I.pb.fits.gz
│   ├── member..J2206-0031_sci.spw27.mfs.I.pbcor.fits
│   ├── member..J2206-0031_sci.spw29.cube.I.mask.fits.gz
│   ├── member..J2206-0031_sci.spw29.cube.I.pb.fits.gz
```

```
|   ├── member..J2206-0031_sci.spw29.cube.I.pbcor.fits
|   ├── member..J2206-0031_sci.spw29.mfs.I.mask.fits.gz
|   ├── member..J2206-0031_sci.spw29.mfs.I.pb.fits.gz
|   ├── member..J2206-0031_sci.spw29.mfs.I.pbcor.fits
|   ├── member..J2206-0031_sci.spw31.cube.I.mask.fits.gz
|   ├── member..J2206-0031_sci.spw31.cube.I.pb.fits.gz
|   ├── member..J2206-0031_sci.spw31.cube.I.pbcor.fits
|   ├── member..J2206-0031_sci.spw31.mfs.I.mask.fits.gz
|   ├── member..J2206-0031_sci.spw31.mfs.I.pb.fits.gz
|   ├── member..J2206-0031_sci.spw31.mfs.I.pbcor.fits
|   ├── member..J2206-0031_sci.spw33.cube.I.mask.fits.gz
|   ├── member..J2206-0031_sci.spw33.cube.I.pb.fits.gz
|   ├── member..J2206-0031_sci.spw33.cube.I.pbcor.fits
|   ├── member..J2206-0031_sci.spw33.mfs.I.mask.fits.gz
|   ├── member..J2206-0031_sci.spw33.mfs.I.pb.fits.gz
|   ├── member..J2206-0031_sci.spw33.mfs.I.pbcor.fits
|   ├── member..J2136p0041_chk.spw25.mfs.I.mask.fits.gz
|   ├── member..J2136p0041_chk.spw25.mfs.I.pb.fits.gz
|   ├── member..J2136p0041_chk.spw25.mfs.I.pbcor.fits
|   ├── member..J2136p0041_chk.spw27.mfs.I.mask.fits.gz
|   ├── member..J2136p0041_chk.spw27.mfs.I.pb.fits.gz
|   ├── member..J2136p0041_chk.spw27.mfs.I.pbcor.fits
|   ├── member..J2136p0041_chk.spw29.mfs.I.mask.fits.gz
|   ├── member..J2136p0041_chk.spw29.mfs.I.pb.fits.gz
|   ├── member..J2136p0041_chk.spw29.mfs.I.pbcor.fits
|   ├── member..J2136p0041_chk.spw31.mfs.I.mask.fits.gz
|   ├── member..J2136p0041_chk.spw31.mfs.I.pb.fits.gz
|   ├── member..J2136p0041_chk.spw31.mfs.I.pbcor.fits
|   ├── member..J2136p0041_chk.spw33.mfs.I.mask.fits.gz
|   ├── member..J2136p0041_chk.spw33.mfs.I.pb.fits.gz
|   └── member..J2136p0041_chk.spw33.mfs.I.pbcor.fits
├── qa
|   ├── member.uid___A001_X3621_X541.imaging_weblog.tgz
|   ├── member.uid___A001_X3621_X541.manual_calimage.weblog.tgz
|   ├── member.uid___A001_X3621_X541.qa2_report.pdf
|   ├── uid___A002_X10f72bf_X1383.qa0_report.pdf
|   ├── uid___A002_X110fd13_X12b88.qa0_report.pdf
|   ├── uid___A002_X110fd13_X12b88__qa2_part*.png
|   ├── uid___A002_X110fd13_X12b88__textfile.txt
|   ├── uid___A002_X110fd13_X131c8.qa0_report.pdf
|   ├── uid___A002_X110fd13_X131c8__qa2_part*.png
|   ├── uid___A002_X110fd13_X131c8__textfile.txt
```

```
|   ├── uid___A002_X110fd13_X1b52b.qa0_report.pdf
|   ├── uid___A002_X110fd13_X1b52b__qa2_part*.png
|   ├── uid___A002_X110fd13_X1b52b__textfile.txt
|   ├── uid___A002_X110fd13_X230eb.qa0_report.pdf
|   ├── uid___A002_X110fd13_X230eb__qa2_part*.png
|   ├── uid___A002_X110fd13_X230eb__textfile.txt
|   ├── uid___A002_X110fd13_X59fd.qa0_report.pdf
|   ├── uid___A002_X110fd13_X59fd__qa2_part*.png
|   └── uid___A002_X110fd13_X59fd__textfile.txt
└── script
    ├── member.uid___A001_X3621_X541.calimage.product_rename.txt
    ├── member.uid___A001_X3621_X541.scriptForImaging.py
    ├── member.uid___A001_X3621_X541.scriptForPI.py
    ├── uid___A002_X110fd13_X12b88.ms.scriptForCalibration.py
    ├── uid___A002_X110fd13_X12b88__calibratorFluxes.xml
    ├── uid___A002_X110fd13_X131c8.ms.scriptForCalibration.py
    ├── uid___A002_X110fd13_X131c8__calibratorFluxes.xml
    ├── uid___A002_X110fd13_X1b52b.ms.scriptForCalibration.py
    ├── uid___A002_X110fd13_X1b52b__calibratorFluxes.xml
    ├── uid___A002_X110fd13_X230eb.ms.scriptForCalibration.py
    ├── uid___A002_X110fd13_X230eb__calibratorFluxes.xml
    ├── uid___A002_X110fd13_X59fd.ms.scriptForCalibration.py
    └── uid___A002_X110fd13_X59fd__calibratorFluxes.xml
```

Note that for simplicity, the names of the files in the *product* directory were shortened here by removing the MOUS UID, which in a real delivery would be contained within the double dots in the names of our example. In the example above, the *qa* directory contains a QA2 report in pdf format. The html format of the QA2 report has been deprecated during Cycle 7. From Cycle 8, the manual weblog is provided in html format and compressed in a tgz file. Because pipeline imaging tasks were used during the imaging process, an * imaging_weblog.tgz is present in the *qa* directory.

This dataset comes from a bandwidth-switching observation with five EBs, each assigned a unique UID. The corresponding calibration scripts in the script directory are named to match each EB UID:

```
    ├── uid___A002_X110fd13_X12b88.ms.scriptForCalibration.py
    ├── uid___A002_X110fd13_X131c8.ms.scriptForCalibration.py
    ├── uid___A002_X110fd13_X1b52b.ms.scriptForCalibration.py
    ├── uid___A002_X110fd13_X230eb.ms.scriptForCalibration.py
    └── uid___A002_X110fd13_X59fd.ms.scriptForCalibration.py
```

In case of full polarization manual calibration, a script for this aim is additionally present and named **.scriptForPolCalibration.py*

Furthermore, if manual imaging was done using CASA task “tclean” directly or indirectly via pipeline tasks, the *script* directory contains the script “**scriptForImaging.py*”.

├— member.uid___A001_X3621_X541.scriptForImaging.py

Finally, like in every delivery, the scriptForPI is included.

├— *member.uid___A001_X3621_X541.scriptForPI.py*

The user is encouraged to inspect all these scripts, as they may also contain helpful comments on special properties of the data.

Unpacking the raw data after downloading should be done at the top level (in the directory containing the directory 2026.1.01573.S in this example). It should result in an additional *raw* directory at the level of the *script* directory:

```
|   |   |   |   |-- raw/
|   |   |   |   |   |-- uid___A002_X110fd13_X12b88.asdm.sdm
|   |   |   |   |   |-- uid___A002_X110fd13_X131c8.asdm.sdm
|   |   |   |   |   |-- uid___A002_X110fd13_X1b52b.asdm.sdm
|   |   |   |   |   |-- uid___A002_X110fd13_X230eb.asdm.sdm
|   |   |   |   |   |-- uid___A002_X110fd13_X59fd.asdm.sdm
```

In order to just reproduce the observatory calibration, the user can then run the “*scriptForPI.py*” by typing at the shell prompt of the operating system:

```
cd script
casa -c "execfile('member.uid___A001_X3621_X541.scriptForPI.py')"
```

Note that the same version of CASA should be used as that of the original processing.

Executing scriptForPI.py will first perform various tests on the directory structure and the presence of the necessary files and then run the corresponding scriptForCalibration.py on each of the ASDMs. If there is more than one ASDM and further steps were necessary in the QA2 process to prepare the dataset for imaging, the scriptForImagingPrep.py will be present and will be run as well.
The results will be placed in a new directory called *calibrated* at the same level as the directories “raw” and “script”:

```
|   |   |   |   |-- calibrated
|   |   |   |   |   |-- uid___A002_ X110fd13_X12b88.calibration
|   |   |   |   |   |-- uid___A002_ X110fd13_X12b88.ms.split.cal
|   |   |   |   |   |-- uid___A002_ X110fd13_X131c8.calibration
|   |   |   |   |   |-- uid___A002_ X110fd13_X131c8.ms.split.cal
|   |   |   |   |   |-- uid___A002_ X110fd13_X1b52b.calibration
|   |   |   |   |   |-- uid___A002_ X110fd13_X1b52b.ms.split.cal
|   |   |   |   |   |-- uid___A002_ X110fd13_X230eb.calibration
|   |   |   |   |   |-- uid___A002_ X110fd13_X230eb.ms.split.cal
|   |   |   |   |   |-- uid___A002_X110fd13_X59fd.calibration
|   |   |   |   |   |-- uid___A002_X110fd13_X59fd.ms.split.cal
```

If the data reducer deems it necessary, the final calibrated visibility data can be concatenated into a single MS called “calibrated.ms” or into separate MSs per observing session (for full-polarization data). Otherwise, the imaging process will operate directly on the MSs for each EB. The intermediate output of CASA to achieve the calibration of each EB is stored in the calibration working directories:

```
|   |   |   |   |   |-- uid___A002_ X110fd13_X12b88.calibration
|   |   |   |   |   |-- uid___A002_ X110fd13_X131c8.calibration
|   |   |   |   |   |-- uid___A002_ X110fd13_X1b52b.calibration
|   |   |   |   |   |-- uid___A002_ X110fd13_X230eb.calibration
|   |   |   |   |   |-- uid___A002_X110fd13_X59fd.calibration
```

while the calibrated visibilities for each EB individually are stored in the MSs with names ending in “.ms.split.cal” (if there is only one EB, this MS will take the role of calibrated.ms).

The execution of scriptForPI.py will produce MSs that are ready for imaging, but will not perform any imaging on them: it will not run scriptForImaging.py.

5.3 Running the *scriptForPI.py* in the case of pipeline-calibrated data

The following shows an example of the contents of a directory tree for a dataset that was calibrated and imaged using the ALMA pipeline. This can easily be seen from the fact that there is a file “***pprequest.xml**” present in the *script* directory, where, like in the case of manually calibrated data, the “*scriptForPI.py*” is found as well.

```
2026.1.00324.L
└── science_goal.uid___A001_X3833_X63f6
   └── group.uid___A001_X3833_X6403
      └── member.uid___A001_X3833_X6404
         ├── calibration
         |   |--  member..hifa_calimage*.auxproducts.tgz
         |   |-- member.uid___A001_X3833_X6404.s1.caltables.tgz
         |   |--  uid___A002_X12f828c_X23370.ms.calapply.txt
         |   |--  uid___A002_X12f828c_X23370.ms.flagversions.tgz
         ├── product
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.cube.I.selfcal.pb.fits.gz
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.cube.I.selfcal.pbcor.fits
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.mfs.I.mask.fits.gz
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.mfs.I.pb.fits.gz
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.mfs.I.pbcor.fits
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.mfs.I.selfcal.mask.fits.gz
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.mfs.I.selfcal.pb.fits.gz
         |   |-- member…_2MASS_J04393364+2359212__sci.spw25.mfs.I.selfcal.pbcor.fits
         |   |-- member…_2MASS_J04393364+2359212__sci.spw27.cube.I.selfcal.mask.fits.gz
         |   |-- member…_2MASS_J04393364+2359212__sci.spw27.cube.I.selfcal.pb.fits.gz
         |   |-- member…_2MASS_J04393364+2359212__sci.spw27.cube.I.selfcal.pbcor.fits
         |   |-- member…_2MASS_J04393364+2359212__sci.spw27.mfs.I.mask.fits.gz
```

| |-- member…_2MASS_J04393364+2359212__sci.spw27.mfs.I.pb.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw27.mfs.I.**pbcor.fits**
| |-- member…_2MASS_J04393364+2359212__sci.spw27.mfs.I.selfcal.mask.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw27.mfs.I.selfcal.pb.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw27.mfs.I.selfcal.pbcor.fits
| |-- member..._J04393364+2359212__sci.spw25_27_29_31_33.cont.I.alpha.error.fits
| |-- member…_2MASS_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.alpha.fits
| |-- member…_2MASS_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.mask.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.pb.tt0.fits
| |-- member…J04393364+2359212__sci.spw25_27_29_31_33.cont.I.selfcal.alpha.error.fits
| |-- member…_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.selfcal.alpha.fits
| |-- member…_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.selfcal.mask.fits.gz
| |-- member…_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.selfcal.pb.tt0.fits
| |-- member…_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.selfcal.tt0.pbcor.fits
| |-- member..._J04393364+2359212__sci.spw25_27_29_31_33.cont.I.selfcal.tt1.pbcor.fits
| |-- member…_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.tt0.pbcor.fits
| |-- member…_J04393364+2359212__sci.spw25_27_29_31_33.cont.I.tt1.pbcor.fits
| |-- member…_2MASS_J04393364+2359212__sci.spw29.cube.I.selfcal.pb.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw29.cube.I.selfcal.pbcor.fits
| |-- member…_2MASS_J04393364+2359212__sci.spw29.mfs.I.mask.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw29.mfs.I.pb.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw29.mfs.I.pbcor.fits
| |-- member…_2MASS_J04393364+2359212__sci.spw29.mfs.I.selfcal.mask.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw29.mfs.I.selfcal.pb.fits.gz
| |-- member…_2MASS_J04393364+2359212__sci.spw29.mfs.I.selfcal.pbcor.fits
| |-- member..._2MASS_J04393364+2359212__sci.spw31.cube.I.selfcal.pb.fits.gz
| |-- member..._2MASS_J04393364+2359212__sci.spw31.cube.I.selfcal.pbcor.fits
| |-- member..._2MASS_J04393364+2359212__sci.spw31.mfs.I.mask.fits.gz
| |-- member..._2MASS_J04393364+2359212__sci.spw31.mfs.I.pb.fits.gz
| |-- member..._2MASS_J04393364+2359212__sci.spw31.mfs.I.pbcor.fits
| |-- member..._2MASS_J04393364+2359212__sci.spw31.mfs.I.selfcal.mask.fits.gz
| |-- member..._2MASS_J04393364+2359212__sci.spw31.mfs.I.selfcal.pb.fits.gz
| |-- member..._2MASS_J04393364+2359212__sci.spw31.mfs.I.selfcal.pbcor.fits
| |-- member.uid___A001_X3833_X6404.J0431+1731_ph.spw25.mfs.I.mask.fits.gz
| |-- member...J0431+1731_**ph**.spw25.mfs.I.pb.fits.gz
| |-- member...J0431+1731_ph.spw31.mfs.I.pbcor.fits
| |-- member...J0431+1731_ph.spw25.mfs.I.mask.fits.gz
| |-- member...J0431+1731_ph.spw25.mfs.I.pb.fits.gz
| |-- member...J0431+1731_ph.spw25.mfs.I.pbcor.fits
| |-- member...J0431+1731_ph.spw27.mfs.I.mask.fits.gz
| |-- member...J0431+1731_ph.spw27.mfs.I.pb.fits.gz
| |-- member...J0431+1731_ph.spw27.mfs.I.pbcor.fits
| |-- member...J0431+1731_ph.spw29.mfs.I.mask.fits.gz
| |-- member...J0431+1731_ph.spw29.mfs.I.pb.fits.gz
| |-- member...J0431+1731_ph.spw29.mfs.I.pbcor.fits

```
|   |-- member...J0510+1800_bp.spw31.mfs.I.mask.fits.gz
|   |-- member...J0510+1800_bp.spw31.mfs.I.pb.fits.gz
|   |-- member...J0510+1800_bp.spw31.mfs.I.pbcor.fits
|   |-- member...J0510+1800_bp.spw25.mfs.I.mask.fits.gz
|   |-- member...J0510+1800_bp.spw25.mfs.I.pb.fits.gz
|   |-- member...J0510+1800_bp.spw25.mfs.I.pbcor.fits
|   |-- member...J0510+1800_bp.spw27.mfs.I.mask.fits.gz
|   |-- member...J0510+1800_bp.spw27.mfs.I.pb.fits.gz
|   |-- member...J0510+1800_bp.spw27.mfs.I.pbcor.fits
|   |-- member...J0510+1800_bp.spw29.mfs.I.mask.fits.gz
|   |-- member...J0510+1800_bp.spw29.mfs.I.pb.fits.gz
|   |-- member...J0510+1800_bp.spw29.mfs.I.pbcor.fits
 ├── qa
|   |--  member..hifa_calimage*.weblog.tgz
|   |--  member..qa2_report.pdf
|   |--  uid___A002_X12f828c_X23370.qa0_report.pdf
 ├── raw
|   |--  uid___A002_ X12f828c_X23370.asdm.sdm
└── script
|-- member...calimage.product_rename.txt
|-- member...hifa_calimage.casa_piperestorescript.py
|-- member...hifa_calimage.casa_pipescript.py
|-- member...hifa_calimage.pipeline_manifest.xml
|-- member...hifa_calimage.pldriver_report.xml
|-- member...hifa_calimage.pprequest.xml
`-- member...scriptForPI.py
```

The file names in the directories “product”, “qa”, and “script” were shortened by not showing the MOUS UID, which is indicated by two consecutive dots in the file name.

As with a manually calibrated dataset, if you want to perform further imaging or re-calibrate, you will first need to download and unpack the raw data. This will create the additional *raw* subdirectory containing the ASDM(s). In order to then run “*scriptForPI.py*”, you will need to verify that you have installed the CASA version indicated in the QA2 Report or Weblog. Then start up CASA in the *script* directory using

```
casa --pipeline
```

At the CASA prompt type

```
execfile('member.uid___A001_X1465_X1b48.scriptForPI.py')
```

Unlike for manually calibrated data, where scriptForPI.py re-runs the calibration by default, for pipeline calibrated data, scriptForPI.py will look for *casa_piperestorescript.py* and execute it. This will

load the flagging and calibration tables included in the archival package and apply them to the raw MSs. The final calibrated MSs will be placed in the *calibrated* directory.

The *casa_piperestorescript.py* can also be executed independently, without requiring *scriptForPI.py* Starting with Cycle 13, the Science Pipeline (version 2026.2.0.27) supports backward-compatible data restoration for observations dating back to Cycle 8. For further details on compatibility with earlier cycles, please refer to the pipeline documentation.

5.4 Running casa_pipescript.py

In the case of pipeline calibrated data, there is an alternative way to generate the calibrated visibilities. It uses the script "*casa_pipescript.py*" that is provided in the *script* directory. This contains the same tasks and parameters that were used when the ALMA QA2 processing calibrated the data. In principle, the user can modify this script to adjust the calibration, but it is recommended to try the delivered version first. The details of the usage of this script are described in the Pipeline documentation (see Section 6).

The script *casa_pipescript.py* can also be run via the scriptForPI.py. Simply move the *casa_piperestorescript.py* out of the *script* directory: scriptForPI will then use *casa_pipescript.py*. For some pipeline-calibrated data (e.g. TP data), the script *casa_piperestorescript.py* may not have been provided. In that case, the scriptForPI.py will run the *casa_pipescript.py* by default.

5.4.1 Considerations for Running Supplemental Manual Imaging Tasks

The user needs to be aware that if supplemental manual imaging was performed, like for any other imaging task, this will not be run by scriptForPI.py, which only deals with the calibration of the dataset. If the user wants to execute the supplemental imaging with different parameters, they will need to edit the scriptForImaging.py found inside the **manual_imaging.tgz** file located in the products directory. Since this is supplemental imaging, it is assumed that the imaging pipeline has been executed beforehand. This scriptForImaging.py runs on the *_target.ms or *_targets.ms and *_targets_line.ms created by the pipeline imaging stage. To be able to run scriptForImaging.py in the case of supplemental imaging, the user will then need to restore the continuum subtraction and create the *_target*.ms. This restore can be triggered by setting CASA variable DOCONTSUB=True before executing scriptForPI.py (see section **Error! Reference source not found.** below) or by following the instructions at this link:
https://casaguides.nrao.edu/index.php?title=ALMA_Imaging_Pipeline_Reprocessing

Once *_target*.ms files have been created, scriptForImaging.py can be run from within CASA. Since the data reducer may have used pipeline tasks to create the supplemental products, it is always better to load CASA with the --pipeline option, to execute scriptForImaging.py.

5.5 Saving disk space during and after the execution of the scriptForPI.py

The ALMA datasets are increasingly large. If the user would like to save disk space and only work with the calibrated data once these were regenerated with "scriptForPI.py", they can move the calibrated MSs of interest from the "calibrated'' directory elsewhere and then delete the calibrated directory.

In case of manually calibrated data, it is also possible to only delete some or all of the intermediate MSs contained in the calibration directories, using

```
cd calibrated; rm -rf *.calibration/*.ms; rm -rf *.calibration/*.ms.split
```

Since the middle of Cycle 1, the scriptForPI.py has also offered the "SPACESAVING" option to limit the disk space usage during and after its execution. In order to make use of this, the Python global variable SPACESAVING needs to be set before starting the script using

```
cd script
casa -c "global SPACESAVING; SPACESAVING=N; execfile('scriptForPI.py')"
```

where N is an integer from 0 to 3 with the following meaning:

SPACESAVING	= 0	same as not set (all intermediate MSs are kept)
	= 1	do not keep intermediate MSs named *.ms.split
	= 2	do not keep intermediate MSs named *.ms and *.ms.split
	>= 3	do not keep intermediate MSs named *.ms, *.ms.split, and *.ms.split.cal (if possible)
	= -1	do not check disk space

With SPACESAVING=0, the required additional diskspace is up to 14 times as large as the delivered data (products and raw data), while with SPACESAVING=3 (maximum savings), it is up to 6 times as large. The script will estimate the required disk space and will not execute if there is not sufficient free space available. You can disable the disk-space checking by setting the value to -1.

5.6 Splitting out the calibrated data at the end of running the scriptForPI.py

For manually calibrated data, it is standard practice to conclude the calibration by copying the calibrated visibilities into a new MS at the end of calibration. This is done using the CASA tasks *mstransform* or *split* and is called "splitting out" the calibrated data. The final calibrated MSs are named *uid*.ms.split.cal.* They contain only the science spectral windows.

For pipeline calibrated and imaged data, the calibrated visibilities for the science target(s) are split to a new MS with **target.ms or *targets.ms* (the latter only from Cycle 9) to facilitate the continuation of the processing with the imaging section of the Science Pipeline. The original MS is called *uid*.ms* and contains all original spectral windows and both the calibrated and uncalibrated data. The calibrated data can be found in the MS table column "CORRECTED". Before Cycle 9, the split MS, called *uid*target.ms* contains two columns of data: the calibrated (continuum + line) data are in the DATA column and the continuum subtracted line visibilities are in the CORRECTED_DATA column ready for line imaging. From Cycle 9 onwards, two split MSs are provided: *uid*targets.ms* and **.targets_line.ms.* The split MS *uid*targets.ms* has the science target calibrated data (continuum + line) in the MS table column DATA which is automatically picked up by the CASA imaging tasks. The split MS named **_targets_line.ms* has the continuum-subtracted calibrated data of the science target in the DATA column. To create the split MS **.targets_line.ms* with the continuum-subtracted line visibilities in the DATA column, the task uvcontsub has to be executed with the imaging pipeline.

From Cycle 10 onwards, these MSs can contain a CORRECTED column with the *self-calibrated* visibilities if self-calibration was successful for the given target.

A summary of the data column contents in each IF calibrated ms file is shown below:

	MS FILENAME	DATA COLUMN	CORRECTED COLUMN
	*ms.split.cal	Calibrated data	N/A
BEFORE CYCLE 9	*target.ms	Calibrated ("cont + line")	Calibrated ("line")
FROM CYCLE 9	*targets.ms	Calibrated ("cont + line")	Self-calibrated *(Cycle 10+)*
	*targets_line.ms	Calibrated ("line")	Self-calibrated (*Cycle 10+)*

If you want to force the *scriptForPI.py* to perform the "splitting out" (to create *uid*.ms.split.cal* for each EB), you can set the variable DOSPLIT to True before starting the script:

casa -c "global SPACESAVING; global DOSPLIT; SPACESAVING=N; DOSPLIT=True; execfile('scriptForPI.py')"

A complete description of the DATA and CORRECTED columns in the MS for IF and TP data pipeline/manually calibrated can be found at:
https://help.almascience.org/kb/articles/what-ms-naming-conventions-does-alma-follow

5.7 Generating the pipeline continuum subtraction products

Starting with data from Cycle 8, it is possible to make the scriptForPI.py also trigger the generation of the continuum subtracted MSs. This is controlled by the variable "DOCONTSUB". Just like with DOSPLIT above, you need to set the variable to True before running the scriptForPI.py using execfile. From Cycle 10 onwards, DOCONTSUB will automatically be run if the PL used self-calibration. You can inhibit the continuum subtraction by setting DOCONTSUB=False.

5.8 Generating plots related to the renormalization task

Since Cycle 7, data suffering from incorrect amplitude normalization caused by bright *astronomical* lines detected in the autocorrelations of some target sources are adjusted when the correction fraction is more than 2%. More information on the subject can be found at the documentation of the ALMA Science Pipeline (see Section 9) and at the following knowledgebase articles:

https://help.almascience.org/kb/articles/what-are-the-amplitude-calibration-issues-caused-by-alma-s-normalization-strategy

https://help.almascience.org/kb/articles/what-errors-could-originate-from-the-correlator-spectral-normalization-and-tsys-calibration

In the pipeline, a comprehensive implementation of this task is available only from Cycle 10 onwards. During Cycle 7, interferometric datasets were checked for the renormalization issue using an external script with CASA 6.1. The diagnostic plots (renorm scaling factors vs. frequency) can be generated by the user employing *scriptForFluxCalibration.py* executed by the scriptForPI.py: the diagnostic plots

are found in the *calibrated/working/Plots* directory. In Cycle 8 and CASA 6.2, the renorm task was included within the pipeline. The diagnostic plots are accessible from the weblog at the hifa_renorm stage clicking the link to diagnostic plots provided in pdf format. Strong atmospheric lines and improper data segmentation may have caused a false triggering of the correction by the pipeline but adjusted manually by the DRs employing an external script. The required commands for a proper renormalization application are incorporated in the *casa_pipescript.py* and the diagnostic plots are reproducible on disk by the user: see *calibrated/working/RN_plots* directory. For full polarization datasets when performing renormalization in the *scriptForPolCalibration.py*, the scriptForPI.py shall be executed with the option SPACESAVING=1 to make sure the MSs are available for a successful renormalization application. During Cycle 9, strong ATM lines are automatically flagged by the pipeline, but false segmentation correction is still improved by the DR with a script using CASA 6.4. The diagnostic plots can be viewed in the weblog under hifa_renorm stage. Also in this case the *casa_pipescript.py* execution produces the diagnostic plots as previously described. Since Cycle 10, no additional manual intervention for the renorm task by the DR is foreseen when employing the pipeline. Diagnostic spectra plots can be inspected in the weblog at the hifa_renorm stage.

5.9 Manually calibrated TP data

If your TP data were reduced manually, you will find scripts "*.*scriptForSDCalibration.py*" in the script folder. If the scriptForPI.py is also present, you can proceed by running it as for other data. For details on manual TP data calibration, please see the official CASA guide available at

https://casaguides.nrao.edu/index.php?title=M100_Band3

https://casadocs.readthedocs.io/en/v6.7.4/notebooks/single_dish_calibration.html

In older deliveries, the scriptForPI.py was not present. In that case, you need to download the raw data (ASDMs) from the ALMA archive, remove the '.asdm.sdm' extensions, and then run the scripts ('*.*scriptForSDcalibration.py*' from the scripts folder) in the same folder with the ASDM.

For the imaging, we recommend you use the provided script, as it contains certain values (e.g. Jy/K conversion factors) that must be set precisely. You could run the script step-by-step, as for the calibration script. Before doing that, you may need to update the paths to the calibrated data, in the msNames variable near the top of the script.

5.10 Pipeline calibrated TP data

Similar to the interferometric pipeline, there are two methods available for the user to restore pipeline calibrated TP data:

(A) The first method requires rerunning the entire pipeline using scriptForPI.py and casa_pipescript.py,

(B) The second method involves using scriptForPI.py and casa_piperestorescript.py with some edits to restore the calibrated MS(s).

For method (A), please refer to Section 5.3.

It is important to note that simply executing scriptForPI.py will NOT create the fully calibrated TP MS(s). The atmospheric correction and the baseline subtraction will still be missing because these two steps are carried out as part of the imaging stage. Method (A) is recommended for users who

are not familiar with the calibration pipeline, as it runs the entire pipeline and creates the calibrated MS(s). However, a disadvantage is that it takes longer than method (B), which restores the first part of the pipeline process using the existing calibration tables. Method (B) is especially useful for big datasets with many EBs.

In the remainder of this section, we explain the procedure for restoring calibrated TP MS(s) using method (B). To restore the calibrated MS(s) using method (B), the user employs the scriptForPI.py and casa_piperestorescript.py files located in the *script* directory. By executing scriptForPI.py, the user runs casa_piperestorescript.py, which restores the calibrated MS(s) from the raw ASDMs by applying the necessary calibration tables. The *casa_piperestorescript.py* can also be executed independently, without requiring *scriptForPI.py.* Starting with Cycle 13, the Science Pipeline (version 2026.2.0.27) supports backward-compatible data restoration for observations dating back to Cycle 8. For further details on compatibility with earlier cycles, please refer to the pipeline documentation. For TP data, casa_piperestorescript.py only restores the TP pipeline tasks from hsd_importdata to hsd_applycal. The calibrated MS(s) will be placed in a newly generated *calibrated* directory. To reproduce the same calibrated MS(s) created by QA2, the user needs to run the remaining pipeline tasks written in casa_pipescript.py (i.e., from hsd_atmcor to the 2nd hsd_blflag). If exactly the same calibrated MS(s) are to be obtained as were used in the QA2 process, then the same CASA version as in QA2 should be used. In case the user wants to process the MS(s) in a different way than QA2 (e.g., skip hsd_atmcor), method (B) is useful.

The CASA guide page below provides some more guidance about how to use ALMA pipeline tasks to restore and modify TP pipeline imaging products. The guide also provides instructions on how to change the applied atmospheric model and how to modify the baseline subtraction.

http://casaguides.nrao.edu/index.php/Single-Dish_(Total_Power)_Data_Processing_with_Pipeline_Cycle_12[3]

[3] An updated version of the guide which includes Cycle 13 SDPL updates is being prepared and will be available soon.

6 Relevant documentation and help

ALMA Quality Assurance is described in chapter 11 of the ALMA Technical Handbook

https://almascience.org/documents-and-tools/cycle13/alma-technical-handbook

For more information on the usage of CASA, please refer to the CASA home page

http://casa.nrao.edu/

Detailed examples are given in the ALMA guides

http://casaguides.nrao.edu/index.php?title=ALMAguides

and in the CASA notebook to run tutorials

https://casadocs.readthedocs.io/en/v6.7.4/notebooks/introduction.html

and the documentation of the ALMA Science Pipeline is available from

https://almascience.org/processing/science-pipeline

If you have any problems with your ALMA data, please contact the ALMA helpdesk at

https://help.almascience.org/

Additional references:

A complete description of the phase and amplitude calibration of ALMA data in VLBI mode can be found in:
Goddi, C., Martí-Vidal, I., Messias, H., et al. 2019, Calibration of ALMA as a Phased Array. ALMA Observations During the 2017 VLBI Campaign, PASP, 131, 075003. **DOI:10.1088/1538-3873/ab136a**

A full description of PolConvert can be found in:
Martí-Vidal, I., Roy, A., Conway, J., and Zensus, A. J. 2016, Calibration of mixed-polarization interferometric observations. Tools for the reduction of interferometric data from elements with linear and circular polarization receivers, A&A, 587, A143. **DOI:10.1051/0004-6361/201526063**



The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of the European Organisation 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 and Technology Council (NSTC) in Taiwan 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.

