

ALMA QA2 Data Products for Cycle 7



www.almascience.org

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
0.1-0.4	January 2014	Dirk Petry
0.5, 1.0	February 2014	Dirk Petry
2.0, 2.1	September 2014	Dirk Petry
2.2	February 2015	Dirk Petry
3.0	November 2015	Dirk Petry
5.0	March 2018	Dirk Petry, Arielle Moullet, Mark Lacy
5.1	May 2018	Dirk Petry, Mark Lacy
7.12	February 2021	Luciano Cerrigone, Dirk Petry

Contributors

Luciano Cerrigone (JAO), Harold Francke (JAO), Misato Fukagawa (NAOJ), John Hibbard (NRAO), Mark Lacy (NRAO), Arielle Moullet (NRAO), Hiroshi Nagai (NAOJ), Kouichiro Nakanishi (NAOJ), Theodoros Nakos (JAO), Dirk Petry (ESO), Tony Remijan (NRAO), Thomas Stanke (ESO), Catarina Ubach (NRAO), and Martin Zwaan (ESO)

In publications, please refer to this document as:

Cerrigone, L. et al., 2021, ALMA QA2 Data Products for Cycle 7, Version 1.0, ALMA

Table of contents

1	Introduction	2
2	Group ObsUnitSet (GroupOUS), Member ObsUnitSet (MemberOUS), and Scheduling Block (SB).....	3
3	Overview	4
4	Data Delivery Products	7
4.1	README	7
4.2	Data Reduction Scripts (directory “script”)	8
4.3	Imaging Products (directory “product”)	9
4.4	CASA Tables (directory “calibration”)	11
4.5	QA Documentation (directory “qa”)	11
5	Generating the calibrated visibilities	12
5.1	Running the scriptForPI.py in the case of manually calibrated data	13
5.2	Running the <i>scriptForPI.py</i> in the case of pipeline-calibrated data	16
5.3	Running casa_pipescript.py	19
5.4	Saving disk space during and after the execution of the scriptForPI.py	20
5.5	Splitting out the calibrated data at the end of running the scriptForPI.py	20
5.6	manually calibrated TP data.....	21
6	Relevant documentation and help	21

1 Introduction

The goal of ALMA Quality Assurance (QA) is to ensure that a reliable final data product is delivered to the PIs and that this has reached the desired control parameters outlined in the science goals, meaning that it is calibrated to the desired accuracy and contains no significant calibration or imaging artefacts. For Cycle 7, as for all previous Cycles, the production of imaging products during QA will be done on a “best effort” basis.

The first level of QA (QA0) is determined shortly after observations, based on metadata and quick-look analysis. The second level of Quality Assurance (QA2) deals with QA after data reduction, where data processing (DP) experts at JAO, the ARCs and/or the ARC nodes inspect the products created by the ALMA pipeline or semi-interactively. This document describes the final science data products that are archived and delivered to the PIs at the end of the QA2 process.

The DP experts perform for each ALMA science dataset a detailed analysis, to confirm that the observations have achieved the science goals requested by the PI¹. In particular, frequency setup, spatial setup, angular resolution, and continuum and/or line detection sensitivity are verified.

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, on the other hand, the requirements are not met, but additional observations of the SB are no longer possible within the cycle time allotted to the project, the data are declared “QA2 semipass” and are also delivered soon after it has become clear that re-observation is not possible (the observing unit has “timed-out”). Data may also be delivered as QA2-semipass if it is determined that the data quality cannot be improved by further observations (for instance, in the case of dynamic-range limited noise), or if all the datasets taken are classified as QA0_semipass and the project has timed out. In the latter case, only the raw data are delivered.

The delivery is made of two parts:

- a) *the science products and supporting material*
- b) *the raw data.*

Part *b*, the raw data, consists of visibility datasets in ALMA native format, the ASDM, which are only needed if the PI would like to perform custom data calibration. On the other hand, 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.

¹ Every dataset is approved by a Data Reduction Manager (DRM) before delivery.

Until September 2014, QA2 work was entirely performed manually using the QA2 calibration script generator tool². 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**”. 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 tool²), we call them as “**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**. See Chapter 13 of the Cycle 7 ALMA Technical Handbook and <https://almascience.org/processing/science-pipeline> for more details on the pipeline.

2 Group ObsUnitSet (GroupOUS), Member ObsUnitSet (MemberOUS), and Scheduling Block (SB)

By design, the smallest scheduling entity which is processed and delivered individually is the Member ObsUnitSet (MemberOUS or MOUS). QA2 therefore is carried out on individual MemberOUSs. As in previous Cycles, the MemberOUS contains also in Cycle 7 essentially always only one Scheduling Block (SB), even in the case of Total Power Array observations.

Hence also in Cycle 7, QA2 continues to be carried out together on all the Execution Blocks (EBs) of a single SB corresponding to one MemberOUS. The products for each MemberOUS are delivered separately, and the delivery of the products for each MemberOUS is labelled with the MemberOUS UID, the unique ALMA archive key.

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

The MemberOUSs are organized into Group ObsUnitSets (in short: GroupOUSs). In the trivial case, there is only one GroupOUS for each Science Goal. In Cycle 7, the following types of MemberOUSs can in principle be in one GroupOUS:

- a) The 12 m Array MemberOUS(s).
- b) The 7 m Array MemberOUS.
- c) The Total Power observation MemberOUS.
- d) The Total Power calibration MemberOUS (may be omitted).

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

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

As of Cycle 7, the archive is offering the possibility to download individual product files. The auxiliary tar file contains calibration tables, logs, scripts, and quality assurance reports, while the tar file marked with 001_of_001 contains the FITS product files of the target and possibly the calibrators. The README file contains a description of what is delivered and how to work with it. Optionally, the user may decide to download also the uncalibrated raw data and these will appear as tar files named

```
<Project ID>_<ExecBlock_UID>_asdm.sdm.tar
```

where ExecBlock_UID indicates the unique ID that identifies the execution block, since one tar file will be available per execution. By untarring any of the tar files in a delivery package, the user will find that a directory tree is built as described in the next section.

3 Overview

An ALMA data delivery in Cycle 7 consists of the seven items shown in the directory tree below. Items 1 to 6 are the main delivery package (part *a*), which should be downloaded first. Item 7 (part *b*) is only required if the users would like to work with the raw visibility data. Once unpacked, all data fall into a standardized directory structure:

```
|-- project_id/  
|  |-- science_goal.ouss_id/  
|  |  |-- group.ouss_id/  
|  |  |  |-- member.ouss_id/
```

				-- README	(1)	
				-- product/	(2)	
				-- calibration/	(3)	
				-- qa/	(4)	
				-- script/	(5)	
				-- log/	(6)	(only present in manually calibrated data)
				-- raw/	(7)	(only present when part <i>b</i> is unpacked)

The directory content is summarized in the following. When an item is marked as “best effort”, this means that it is generated by the data reducers only if sufficient resources are available and it is compatible with the science goal.

1) README

A text file explaining where to find information about the QA2 results. The main source of information on the data processing is now the “weblog”, item (4), in the subdirectory “qa”.

2) Imaging Products (in FITS format) in subdirectory “product”

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 is what is called “mitigation” of the data products. See the pipeline documentation at <https://almascience.org/processing/science-pipeline>.

b) for pipeline-calibrated data:

- Continuum images of the bandpass and phase calibrators and (if present) the check source(s), for each of the science spectral windows.

c) manually imaged line cubes:

- For the representative spectral window: One spectral cube of “representative channels” for at least the representative target in the MemberOUS, 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.

- (best effort) Continuum image of all non-edge, non-line channels.

d) manually imaged continuum and or aggregate bandwidth images:

- One image of all continuum spectral windows, all non-edge channels.
- (best effort) One image of all continuum spectral windows, all non-line channels.

3) CASA Tables (in CASA Table format) in subdirectory “calibration”

a) for pipeline-calibrated data

- Calibration tables (Antenna Position, Tsys, WVR, Bandpass, Gain, Amplitude).
- Flagversions tables.
- Calibrator fluxes from the ALMA calibration database (flux.csv).
- Antenna positions (antennapos.csv).
- List of the applycal commands for the delivered calibration table (*calapply.txt).
- Flagging commands added manually by the pipeline operator (*flagtemplate.txt and/or *flagtargetstemplate.txt).
- auxcaltables.tgz, auxcalapply.txt.
- hifa_calimage.auxproducts.tgz
- cont.dat.

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

b) for manually calibrated data

- Calibration tables (Tsys, WVR, Bandpass, Gain, Amplitude).
- Flagversions tables.
- Calibration plots and wvrgcal diagnostic output.

4) QA documentation in subdirectory “qa”

a) for pipeline-calibrated data

- The Pipeline Weblog – 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 and flag them. The weblog also contains the log of the commands executed by the pipeline.

- A QA0 report for each execution block. 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 in both html and pdf format. This contains information about the quality assessment performed after processing the data and the comments from the reviewer who decided to deliver the dataset.

b) for manually calibrated data

- QA Reports for all the imaged data (png images and txt files).

5) Data Reduction Scripts (ASCII files) in subdirectory “script”

a) for pipeline-calibrated data

- The Python scripts needed to restore the calibrated MS(s) or rerun the entire pipeline.
- The pipeline processing request file (PPR*.xml or *pprequest.xml).
- The imaging preparation script (if necessary) and the imaging script (may be a dummy).

b) for manually calibrated data

- CASA reduction scripts including: calibration scripts, imaging preparation script (if necessary), imaging script, and script to set up and run the calibration.

6) calibration and imaging log files in subdirectory “log”³

- CASA log files from the QA2 processing.

7) Datasets (in ASDM format) in subdirectory “raw” (if downloaded)

- (for optional download) The Execution Blocks (EBs) that were used in the imaging/QA2 assessment.
- (for optional download if applicable) Additional datasets that were NOT used in the data reduction but may still contain valuable scientific information.

³ This directory is present only for manually calibrated or manually imaged datasets.

4 Data Delivery Products

4.1 README

The complete name of this file in normal deliveries is <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 has normally been just a reference to where one can find additional information.

4.2 Data Reduction Scripts (directory “script”)

In the subdirectory “script”, the user finds the CASA data reduction scripts (Python) which were used to calibrate and image the data. Depending on whether the Science Pipeline was used or the data were calibrated manually, the content of this directory varies:

- 1) *uid....ms.scriptForCalibration.py*: one calibration script for each Execution Block of manually calibrated interferometric data.
- 2) *uid....ms.scriptForSDCalibration.py*: one calibration script for each Execution Block of manually calibrated single-dish data. See section 5.
- 3) *scriptForImagingPrep.py* (for manually imaged datasets, if needed): This script contains all necessary steps to prepare imaging that needs to be performed on all calibrated EBs together (therefore cannot be performed by the calibration script for the individual EBs). These steps may include alignment of frequency grids before concatenation, the concatenation into one MeasurementSet (MS), adjustments to the flux calibration etc. The result is a MS or a set of MSs that are ready for the 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): special calibration routine for full-polarization data that is executed after the per-EB-calibration scripts. 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 next section) as those that are stored in the “products” subdirectory of the delivery. Where convenient for the data reducer, the *scriptForImaging.py* may also contain elements that could in principle be put into a *scriptForImagingPrep.py*. In that case, a *scriptForImagingPrep.py* is not created.
- 6) *scriptForSDImaging.py*: (for manually imaged single-dish data) this script generates the same imaging products (see next section) as those that are stored in the “products” subdirectory of the delivery. See also section 5.
- 7) ***scriptForPI.py***: in all cases, whether pipeline- or manually calibrated, run this script in the “script” directory in order to regenerate the calibrated MeasurementSet (MS) of the

delivered data, after downloading the raw ASDMs. The README file contains instructions (or points to them) on how to do this. Running the *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 section 5 for more details.

Until mid-Cycle 5, the only type of dataset that was *not* delivered with a *scriptForPI.py* was the *manually calibrated Total Power* data. See section 5.

- 8) *casa_piperestorescript.py*: this script is used by *scriptForPI.py* to restore the calibrated MS(s) from the raw ASDMs by applying the necessary calibration tables. This is only provided in the case of pipeline calibration. If it is not provided or it has been removed, the *scriptForPI.py* will run the *casa_pipescript.py* instead.
- 9) *casa_pipescript.py*: this script performs the calibration from scratch using Pipeline tasks. See the Pipeline documentation (see section 6) for more details.
- 10) The pipeline processing request (PPR): an XML file that was used to drive the Pipeline calibration. See <https://almascience.org/processing/science-pipeline> .

Furthermore, there may be the following script file present in case of solar observations: *sun_reduction_util.py* (contains helper routines specific to solar data analysis).

4.3 Imaging Products (directory “product”)

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*, located in the *script* directory. These imaging products were created from the calibrated dataset produced when the *scriptForPI.py* is run. For pipeline-calibrated interferometric data, this directory also contains calibrator images generated by the Pipeline.

A coherent **file naming scheme** was introduced for all imaging products:

<MOUS>.<target>_<target type>.spw<SPWs>.<imagetype>.<stokes>.<comment>.<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 an aggregate bandwidth image, “cube” for a spectrally resolved image, “cont” for an image with the line channels excluded (in pipeline calibrated/imaged data, this will have all spectral windows merged);

<stokes> is the name of the image Stokes parameter (e.g. “I”);

<comment> can be an explanatory text, such as a molecular transition;

<x> is “pbcpr” for the primary beam corrected image, “pb” for the primary beam itself, “mask” for the mask image, “sd” for single dish;

<ext> is the file name extension describing the file type, e.g. “fits”.

Note that the <comment> 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/index.php?/Knowledgebase/Article/View/375> .

The goal of ALMA QA2 in Cycle 7 remains that the calibration is reliable and "science ready", but the imaging products may not be. A sufficient fraction of the possible imaging products is 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).

Care has been taken that the images conform to the FITS standard 3.0.

Where necessary (i.e. where a clean mask was actually applied), the imaging masks are provided as CASA images. These represent the final mask used in the last cleaning iteration. The masks used in previous iterations may have been different (smaller).

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

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

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

For **Total Power (TP)** data, images without primary beam correction are delivered. Also, images of calibrators are not included (as the beam calibrator often is part of a separate observation).

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 *products* directory a file named *manual_imaging.tgz* containing the CASA log files and the *scripForImaging.py* used to perform such supplemental imaging. The user should take care not to mix up this *scripForImaging.py* with that in the *script* directory, which is related to the pipeline run. 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 CASA Tables (directory “calibration”)

In order to further document the process of data calibration, 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) on how to do that. However, many of the plots of interest in this context are also available in the QA documentation (see next section).

In the case of pipeline-calibrated data, the files in this directory are needed to restore the calibrated MS with “*casa_piperestorescript.py*” or, in absence of the latter, with “*casa_pipescript.py*”, which in turn 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 also in this case available in the Pipeline Weblog, which can be found in the directory “qa” (see next section).

4.5 QA Documentation (directory “qa”)

In the subdirectory “qa” 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 *weblog.tar.gz
```

in the “qa” directory and then use your favourite web browser (firefox recommended) to open the resulting file

pipeline*/html/index.html

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). 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.
- 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) Phase 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).

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.

Based on the calibration tables in the “calibration” directory, the user can produce additional diagnostic plots in CASA using the tasks *plotms*, *plotcal*, 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 MeasurementSet (MS) from the raw data (see the next section). Once this is done, the CASA task *plotms* can be used to generate the plots.

5 Generating the calibrated visibilities

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

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.

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` in the case of manually calibrated data

For manually calibrated data, the `scriptForPI` will find the scripts named `scriptForCalibration.py` and `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 essentially the same final product, although the naming differs.

2019.1.01482.T

```
├── science_goal.uid__A001_X1465_Xb46
│   ├── group.uid__A001_X1465_Xb4a
│   │   └── member.uid__A001_X1465_Xb4b
│   │       ├── calibration
│   │       │   ├── member.uid__A001_X1465_Xb4b.concat_S1.ms.TDM.Df0.plots.tgz
│   │       │   ├── uid__A002_Xe5aacf_X29ee.calibration.plots.tgz
│   │       │   ├── uid__A002_Xe5aacf_X29ee.calibration.tgz
│   │       │   ├── uid__A002_Xe5aacf_X29ee.ms.wvrgcal.txt
│   │       │   ├── uid__A002_Xe5aacf_X2d6f.calibration.plots.tgz
│   │       │   ├── uid__A002_Xe5aacf_X2d6f.calibration.tgz
│   │       │   └── uid__A002_Xe5aacf_X2d6f.ms.wvrgcal.txt
```

- ├── uid__A002_Xe5aacf_X3155.calibration.plots.tgz
- ├── uid__A002_Xe5aacf_X3155.calibration.tgz
- ├── uid__A002_Xe5aacf_X3155.ms.wvrgcal.txt
- ├── log
 - ├── member.uid__A001_X1465_Xb4b.analysis.log.tgz
 - ├── member.uid__A001_X1465_Xb4b.calibrated.log.tgz
 - ├── uid__A002_Xe5aacf_X29ee.log.tgz
 - ├── uid__A002_Xe5aacf_X2d6f.log.tgz
 - └── uid__A002_Xe5aacf_X3155.log.tgz
- ├── member.uid__A001_X1465_Xb4b.README.txt
- ├── product
 - ├── member..concat_S1.ms.TDM_img1.mask.tgz
 - ├── member..GRB_pol1_sci.spw0_1_2_3.mfs.A.manual.pbcor.fits
 - ├── member..GRB_pol1_sci.spw0_1_2_3.mfs.IQUV.manual.mask.tgz
 - ├── member..GRB_pol1_sci.spw0_1_2_3.mfs.IQUV.manual.pbcor.fits
 - ├── member..GRB_pol1_sci.spw0_1_2_3.mfs.IQUV.manual.pb.fits.gz
 - ├── member..GRB_pol1_sci.spw0_1_2_3.mfs.P.manual.pbcor.fits
 - ├── member..J1058p0133_polleak.spw0_1_2_3.mfs.A.manual.pbcor.fits
 - ├── member..J1058p0133_polleak.spw0_1_2_3.mfs.IQUV.manual.mask.tgz
 - ├── member..J1058p0133_polleak.spw0_1_2_3.mfs.IQUV.manual.pbcor.fits
 - ├── member..J1058p0133_polleak.spw0_1_2_3.mfs.IQUV.manual.pb.fits.gz
 - ├── member..J1058p0133_polleak.spw0_1_2_3.mfs.P.manual.pbcor.fits
 - ├── member..target_check.mask.tgz
 - ├── uid__A002_Xe5aacf_X29ee__target_check.mask.tgz
 - ├── uid__A002_Xe5aacf_X2d6f__target_check.mask.tgz
 - └── uid__A002_Xe5aacf_X3155__target_check.mask.tgz
- ├── qa
 - ├── member..concat_S1.ms.TDM.Dterm_sky_imag_spw0-1.png
 - ├── member..concat_S1.ms.TDM.Dterm_sky_imag_spw2-3.png
 - ├── member..concat_S1.ms.TDM.Dterm_sky_real_spw0-1.png
 - ├── member..concat_S1.ms.TDM.Dterm_sky_real_spw2-3.png
 - ├── member..concat_S1.ms.TDM.GainRatiosPol.png
 - ├── member..concat_S1.ms.TDM.Gxyamp.png
 - ├── member..concat_S1.ms.TDM.GxyampRatio.png
 - ├── member..concat_S1.ms.TDM.XY0_Amb-NoAmb.png
 - ├── member..concat_S1.ms.TDM.XY-CrossPhase.png
 - ├── member.uid__A001_X1465_Xb4b.qa2_report.html
 - ├── member.uid__A001_X1465_Xb4b.qa2_report.pdf
 - ├── member.uid__A001_X1465_Xb4b.weblog.pdf
 - ├── uid__A002_Xe5aacf_X29ee.qa0_report.pdf
 - ├── uid__A002_Xe5aacf_X29ee_qa2_part1.png
 - ├── uid__A002_Xe5aacf_X29ee_qa2_part2.png
 - ├── uid__A002_Xe5aacf_X29ee_qa2_part3.png
 - ├── uid__A002_Xe5aacf_X29ee_textfile.txt
 - └── uid__A002_Xe5aacf_X2d6f.qa0_report.pdf

```

|   ├── uid__A002_Xe5aacf_X2d6f__qa2_part1.png
|   ├── uid__A002_Xe5aacf_X2d6f__qa2_part2.png
|   ├── uid__A002_Xe5aacf_X2d6f__qa2_part3.png
|   ├── uid__A002_Xe5aacf_X2d6f__textfile.txt
|   ├── uid__A002_Xe5aacf_X3155.qa0_report.pdf
|   ├── uid__A002_Xe5aacf_X3155__qa2_part1.png
|   ├── uid__A002_Xe5aacf_X3155__qa2_part2.png
|   ├── uid__A002_Xe5aacf_X3155__qa2_part3.png
|   └── uid__A002_Xe5aacf_X3155__textfile.txt
└── script
    ├── member.uid__A001_X1465_Xb4b.calimage.product_rename.txt
    ├── member.uid__A001_X1465_Xb4b.scriptForImaging.py
    ├── member.uid__A001_X1465_Xb4b.scriptForPI.py
    ├── member.uid__A001_X1465_Xb4b.scriptForPolCalibration.py
    ├── uid__A002_Xe5aacf_X29ee.ms.scriptForCalibration.py
    ├── uid__A002_Xe5aacf_X2d6f.ms.scriptForCalibration.py
    └── uid__A002_Xe5aacf_X3155.ms.scriptForCalibration.py

```

Note that for simplicity, the names of the files in the “product” and “qa” directory were shortened here by removing the MOUS ID, which in a real delivery would be contained within the double dots in the names of our example.

This dataset is from a polarization observation and has three Execution Blocks (EBs) with UIDs uid__A002_Xe5aacf_X29ee, uid__A002_Xe5aacf_X2d6f, and uid__A002_Xe5aacf_X3155. The directory “script” therefore has three scripts for calibration:

```

├── uid__A002_Xe5aacf_X29ee.ms.scriptForCalibration.py
├── uid__A002_Xe5aacf_X2d6f.ms.scriptForCalibration.py
└── uid__A002_Xe5aacf_X3155.ms.scriptForCalibration.py

```

and, since additional commands are necessary to perform the polarization calibration, a script for this aim is also present:

```

├── member.uid__A001_X1465_Xb4b.scriptForPolCalibration.py

```

Furthermore, like in every delivery, the directory “script” contains the scripts “*scriptForImaging.py” and “*scriptForPI.py”.

```

├── member.uid__A001_X1465_Xb4b.scriptForImaging.py
├── member.uid__A001_X1465_Xb4b.scriptForPI.py

```

The user could have also found a *scriptForImagingPrep.py*, which is an optional script sometimes used by a data reducer to perform further adjustments (if deemed necessary) after the calibration, for example flux equalization.

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 2019.1.01482.T in this example). It should result in an additional directory “raw” at the level of the directory “script”:

```
| | | | |-- raw/  
| | | | |-- uid___A002_Xe5aacf_X29ee.asdm.sdm  
| | | | |-- uid___A002_Xe5aacf_X2d6f.asdm.sdm  
| | | | |-- uid___A002_Xe5aacf_X3155.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_X1465_Xb4b.scriptForPI.py')"
```

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

Executing *scriptForPI* will first perform various tests on the directory structure and the presence of the necessary files and then run the corresponding *scriptForCalibration* 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* 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  
| | | | |-- concat_S1.ms  
| | | | |-- uid___A002_Xe5aacf_X29ee.calibration  
| | | | |-- uid___A002_Xe5aacf_X29ee.ms.split.cal  
| | | | |-- uid___A002_Xe5aacf_X2d6f.calibration  
| | | | |-- uid___A002_Xe5aacf_X2d6f.ms.split.cal  
| | | | |-- uid___A002_Xe5aacf_X3155.calibration  
| | | | |-- uid___A002_Xe5aacf_X3155.ms.split.cal
```

If the data reducer deems it necessary, the final calibrated visibility data combined for all EBs can be found in a MS called “calibrated.ms” or, like in our example, “concat_S1.ms”, where S1 is due to the polarization nature of the dataset and indicates Session 1. The concatenated MS can also not be present, if the data reducer does not deem it necessary to perform a concatenation, and the imaging 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_Xe5aacf_X29ee.calibration
| | | | | |-- uid__A002_Xe5aacf_X2d6f.calibration
| | | | | |-- uid__A002_Xe5aacf_X3155.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`.

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

The following shows the contents of the 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. Like in the case of manually calibrated data, the “*scriptForPI.py*” is found in the “script” directory.

2019.1.01056.S

```
└─ science_goal.uid__A001_X1465_X1b46
  └─ group.uid__A001_X1465_X1b47
    └─ member.uid__A001_X1465_X1b48
      ├── calibration
      │   ├── member..hifa_calimage.auxproducts.tgz
      │   ├── member.uid__A001_X1465_X1b48.session_1.auxcaltables.tgz
      │   ├── member.uid__A001_X1465_X1b48.session_1.caltables.tgz
      │   ├── member.uid__A001_X1465_X1b48.session_2.auxcaltables.tgz
      │   ├── member.uid__A001_X1465_X1b48.session_2.caltables.tgz
      │   ├── uid__A002_Xe1baa0_X5b8c.ms.calapply.txt
      │   ├── uid__A002_Xe1baa0_X5b8c.ms.flagversions.tgz
      │   ├── uid__A002_Xe1baa0_X5b8c_target.ms.auxcalapply.txt
      │   ├── uid__A002_Xe1d2cb_X31f.ms.calapply.txt
      │   ├── uid__A002_Xe1d2cb_X31f.ms.flagversions.tgz
      │   └─ uid__A002_Xe1d2cb_X31f_target.ms.auxcalapply.txt
      └─ product
```

| |— member.._G24.73p0.69__sci.spw16_18_20_22.cont.l.mask.fits.gz
| |— member.._G24.73p0.69__sci.spw16_18_20_22.cont.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw16_18_20_22.cont.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw16.cube.l.mask.fits.gz
| |— member.._G24.73p0.69__sci.spw16.cube.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw16.cube.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw16.mfs.l.mask.fits.gz
| |— member.._G24.73p0.69__sci.spw16.mfs.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw16.mfs.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw18.cube.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw18.cube.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw18.mfs.l.mask.fits.gz
| |— member.._G24.73p0.69__sci.spw18.mfs.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw18.mfs.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw20.cube.l.mask.fits.gz
| |— member.._G24.73p0.69__sci.spw20.cube.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw20.cube.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw20.mfs.l.mask.fits.gz
| |— member.._G24.73p0.69__sci.spw20.mfs.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw20.mfs.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw22.cube.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw22.cube.l.pb.fits.gz
| |— member.._G24.73p0.69__sci.spw22.mfs.l.mask.fits.gz
| |— member.._G24.73p0.69__sci.spw22.mfs.l.pbcor.fits
| |— member.._G24.73p0.69__sci.spw22.mfs.l.pb.fits.gz
| |— member..J1832-1035_ph.spw16.mfs.l.mask.fits.gz
| |— member..J1832-1035_ph.spw16.mfs.l.pbcor.fits
| |— member..J1832-1035_ph.spw16.mfs.l.pb.fits.gz
| |— member..J1832-1035_ph.spw18.mfs.l.mask.fits.gz
| |— member..J1832-1035_ph.spw18.mfs.l.pbcor.fits
| |— member..J1832-1035_ph.spw18.mfs.l.pb.fits.gz
| |— member..J1832-1035_ph.spw20.mfs.l.pbcor.fits
| |— member..J1832-1035_ph.spw20.mfs.l.pb.fits.gz
| |— member..J1832-1035_ph.spw22.mfs.l.mask.fits.gz
| |— member..J1832-1035_ph.spw22.mfs.l.pbcor.fits
| |— member..J1832-1035_ph.spw22.mfs.l.pb.fits.gz
| |— member..J1912-0804_ph.spw16.mfs.l.mask.fits.gz
| |— member..J1912-0804_ph.spw16.mfs.l.pbcor.fits
| |— member..J1912-0804_ph.spw16.mfs.l.pb.fits.gz

```

|   ├── member..J1912-0804_ph.spw18.mfs.l.mask.fits.gz
|   ├── member..J1912-0804_ph.spw18.mfs.l.pbcor.fits
|   ├── member..J1912-0804_ph.spw18.mfs.l.pb.fits.gz
|   ├── member..J1912-0804_ph.spw20.mfs.l.mask.fits.gz
|   ├── member..J1912-0804_ph.spw20.mfs.l.pbcor.fits
|   ├── member..J1912-0804_ph.spw20.mfs.l.pb.fits.gz
|   ├── member..J1912-0804_ph.spw22.mfs.l.mask.fits.gz
|   ├── member..J1912-0804_ph.spw22.mfs.l.pbcor.fits
|   ├── member..J1912-0804_ph.spw22.mfs.l.pb.fits.gz
|   ├── member..J1924-2914_bp.spw16.mfs.l.pbcor.fits
|   ├── member..J1924-2914_bp.spw16.mfs.l.pb.fits.gz
|   ├── member..J1924-2914_bp.spw18.mfs.l.pbcor.fits
|   ├── member..J1924-2914_bp.spw18.mfs.l.pb.fits.gz
|   ├── member..J1924-2914_bp.spw20.mfs.l.pbcor.fits
|   ├── member..J1924-2914_bp.spw20.mfs.l.pb.fits.gz
|   ├── member..J1924-2914_bp.spw22.mfs.l.pbcor.fits
|   └── member..J1924-2914_bp.spw22.mfs.l.pb.fits.gz
└── qa
    ├── member..hifa_calimage.weblog.tgz
    ├── member..qa2_report.html
    ├── member..qa2_report.pdf
    ├── uid__A002_Xe1baa0_X5b8c.qa0_report.pdf
    └── uid__A002_Xe1d2cb_X31f.qa0_report.pdf
└── raw
    ├── uid__A002_Xe1baa0_X5b8c.asdm.sdm
    └── uid__A002_Xe1d2cb_X31f.asdm.sdm
└── script
    ├── member..calimage.product_rename.txt
    ├── member..hifa_calimage.casa_commands.log
    ├── member..hifa_calimage.casa_piperestorescript.py
    ├── member..hifa_calimage.casa_pipescript.py
    ├── member..hifa_calimage.pipeline_manifest.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. Note the additional calibrator images created by the Science Pipeline.

Also here, as for a manually calibrated dataset, you will first need to download and unpack the raw data, if you want to perform further imaging or re-calibrate. This will create the additional subdirectory “raw” 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 working directory “script” using

```
casa --pipeline
```

At the CASA prompt then 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 actually look for *casa_piperestorescript.py* and execute it, if found. This will load the flagging and calibration tables delivered by the Data Reduction Manager (DRM) and apply them to the raw MSs. The final calibrated MSs will be placed in the directory “calibrated”.

The *casa_piperestorescript.py* can also be run stand-alone, without the *scriptForPI*. See the pipeline documentation for more details.

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 again 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, therefore this *scriptForImaging.py* runs on the **_target.ms* created by the imaging pipeline and cannot be executed right away after *scriptForPI.py* has finished. The **_target.ms* contains two columns of data: in the DATA column one finds the calibrated data of the science target and in the CORRECTED column the continuum-subtracted calibrated data. To be able to run *scriptForImaging.py* in the case of supplemental imaging, the user will then need first to restore the continuum subtraction and create the **_target.ms*, which can be done following the instructions at this link [CASA PL Guide](#) (see section Restore Pipeline Continuum Subtraction and Manually Make Image Products). Once **_target.ms* has 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*.

5.3 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 DRM 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 below).

The script *casa_pipescript.py* can also be run via the *scriptForPI*. 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. Total Power data), the script *casa_piperestorescript.py* may not have been provided. In that case, the *scriptForPI* will run the *casa_pipescript.py* by default.

5.4 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 either delete all calibration directories (`cd calibrated; rm -rf *.calibration`) or, in case of manually calibrated data, 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 the run its execution. In order to make use of this, the Python global variable *SPACESAVING* needs to be set before starting the script, e.g. using

```
cd script
casa -c "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 disk space 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.

5.5 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, this step is by default omitted in order to facilitate the continuation of the processing with the imaging section of the Science Pipeline. So the final 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_DATA”, which is automatically picked up by the CASA imaging tasks.

If you want to force the `scriptForPI.py` to perform the “splitting out”, i.e. to create `uid*.ms.split.cal` for each EB, you can set the variable `DOSPLIT` to `True` before starting the script:

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

5.6 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 .

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. beams) that must be set to a precise value. 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.

6 Relevant documentation and help

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

<https://almascience.org/documents-and-tools/cycle7/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 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/>



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.

