ALMA QA2 Data Products for Cycle 1

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

Contributors

Paola Andreani (ESO), John Hibbard (NRAO), Shinya Komugi (NAOJ), Dirk Petry (ESO), Scott Schnee (NRAO), Ken Tatematsu (NAOJ), Baltasar Vila Vilaro (JAO), Eric Villard (JAO)

In publications, please refer to this document as:

Petry, D. et al., 2014, ALMA QA2 Data Products for Cycle 1, Version 1.0, ALMA

Table of contents

1 Introduction

The ALMA QA2 data reduction team performs for each ALMA science data set a detailed analysis to confirm that the observations have achieved the characteristics requested by the PI. In particular, frequency setup, spatial setup, and continuum and line detection sensitivity are verified.

If the requirements are met, the data are declared “QA2 pass”, packaged in a standardized way, and delivered to the PI. The delivery is made in two parts:

(a) the science products and supporting material

(b) (optionally) the raw data.

Part (b), the raw data consist of visibility datasets in ALMA's native format, the ASDM, which are only needed if the PI would like to perform custom data calibration and imaging. Part (a) of the delivery on the other hand, the “QA2 Products”, consists of the QA2 data reduction scripts, log files, calibration 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.

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

By design, the smallest scheduling entity which should be processed individually is the Member ObsUnitSet (MemberOUS or Member). QA2 therefore is carried out on individual Members.

In Cycle 1, the Member contains in almost all cases only one Scheduling Block (SB), even in the case of Total Power Array observations.

Hence, in Cycle 1, QA2 is carried out together on all the Execution Blocks (EBs) of a single SB corresponding to one Member. The products for each Member are delivered separately, and the delivery of the products for each Member is labeled with the Member UID.

The Members are organized into Group ObsUnitSets (in short: Groups). In the trivial case there is only one Group for each Science Goal. In Cycle 1, at most four Members can be in one Group:

a) The 12 m Array Member

b) The Total Power observation Member

c) The Total Power calibration Member (may be omitted)

d) The 7 m Array Member

Once all Members of a Group have passed QA2, another QA2 step has to take place which performs the combined imaging of the up to three elements: 12 m Array, Total

Power Array, and 7 m Array. The delivery of the resulting products will be labeled with the Group UID.

The delivery packaging results in a tar file which is named in the following way:

<Project ID>_<MemberOUS_UID>_m_of_n.tar

for each member where n is the total number of tar files into which the delivery is split (for faster transfer) and m is the number of the individual tar file (typically both m and n are 1 since these files are usually small).

Group deliveries only take place for the case there is more than one Member in the Group and in that case the PI will receive the products for each Member separately and then one more delivery for the combination products. The latter will be in a tar file named

<Project ID>_<GroupOUS_UID>_m_of_n.tar

3 Overview

An ALMA data delivery consists of the following items. Items 1 to 4 are delivered in one package (part (a)) which should be downloaded first. Item 5 (part (b)) is available for download separately and is only required if the users would like to work with the visibility data. Once unpacked, all data falls into a standardized directory structure:

```
|-- project_id/
|   |-- sg_ouss_id/
|   |   |-- group_ouss_id/
|   |   |   |-- member_ouss_id/
|   |   |   |   |-- README
|   |   |   |   |-- script/
|   |   |   |   |-- calibration/
|   |   |   |   |-- qa/
|   |   |   |   |-- product/
|   |   |   |   |-- log/
|   |   |   |   |-- raw/  (only present when part (b) was unpacked)
```

The directory contents is summarized in the following.

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

a) Line:

— For each spectral window: One spectral cube of "representative channels" for at least one target in the MemberOUS (Member), made at the requested resolution and the relevant spectral resolution considering the bandwidth specifed for sensitivity. Coarse continuum subtraction is applied for sources with bright continuum. This applies both to deliveries of single Members and to combined datasets

— (best effort) For each source and each spectral window: cubes of all species specifically listed in the proposal, made at full spatial resolution and the relevant spectral resolution and including ~10% line-free channels on either side. Continuum subtraction is applied for sources with bright continuum. A note will be added if additional species are found in the data, but imaging may not be carried out.

— (best effort) For the same source with multiple non-continuum spectral windows: line cubes should be made with constant velocity channels sampled on the same grid and covering the same velocity range.

— (best effort) Zeroth moment of each line cube

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

b) Continuum:

— One image per sideband, all non-edge channels

— (best effort) One image per sideband, all non-line channels

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

— Tsys and WVR tables

— Flagging tables

— Calibration tables (Bandpass, Gain, Amplitude)

3) QA documentation (png images and pdf files) in subdirectory "qa"

— QA Reports for all the imaged data.

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

— README file summarizing the QA2 results and explaining structure of tarball and file naming conventions

— CASA reduction scripts including: calibration scripts, flux equalization script (if necessary), imaging script, and script to apply calibrations.

— Reduction logs, CASA outputs

— (if necessary) Additional support images/plots as needed

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

— (for optional download) the ExecBlocks 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

4 Data Delivery Products

4.1 README

The first item the user should inspect after unpacking the delivery tar file (part (a)), is the README file. It contains a summary of the QA2 results, in particular the achieved beam and image noise RMS.

4.2 Data Reduction Scripts (directory “script”)

In the subdirectory “script”, the user then finds the CASA data reduction scripts (Python) which were used to calibrate and image the data:

1) uid....ms.scriptForCalibration.py: one calibration script for each Execution Block

2) scriptForFluxCalibration.py (if necessary): if there was more than one Execution Block, and they were observed within one day from each other, a combined flux calibration is done assuming that the flux calibrator had the same intensity during all executions. This script performs this procedure. The result is a combined dataset of all executions.

3) scriptForImaging.py: this script regenerates the imaging products (see next section) as they are stored in the “products” subdirectory of the delivery.

4) scriptForPI.py: run this script in the “script” directory in order to regenerate the calibrated MeasurementSet of the delivered data. The README file contains instructions on how to do this. Running the scriptForPI.py requires that you have downloaded the ASDM datasets and unpacked them such that they reside in the “raw” directory (created during unpacking). See Section 5 for more details.

4.3 Imaging Products (directory “product”)

The “product” subdirectory of the delivery contains the imaging products that were generated with the script “scriptForImaging.py” from the “script” subdirectory based on the calibrated dataset which is obtained when the “scriptForPI.py” is run.

The general idea is that PIs who do not have special requirements on calibration or imaging, can use the FITS images in the products directory to directly proceed with their scientific work and use these images in their publications. *However, ALMA is not yet guaranteeing science-grade imaging products and the PI is encouraged to scrutinize the products thoroughly.* The CASA viewer software can be used to turn the spectral cubes and continuum images into publication-ready plot. But the PI can also use any other image processing package which reads FITS files.

Care has been taken that the images conform with the latest FITS standard (3.0).

Where necessary (i.e. where they cannot be supplied analytically within the scriptForImaging.py text), the imaging masks are provided as CASA images.

4.4 CASA Tables (directory "calibration")

In order to document further the process of data calibration, all calibration tables generated by the calibration scripts are stored in the "calibration" subdirectory of the delivery. How each table was created can be looked up in the scriptForCalibration.py and the scriptForFluxCalibration.py scripts. CASA provides functionality to view these tables. Please refer to the CASA documentation (see Section 10) on how to do that. Many of the plots of interest in this context are, however, also already available in the QA documentation (see next section).

4.5 QA Documentation (directory "qa")

In the subdirectory "qa" of the delivery, the user finds for each execution block and labeled with the execution block UID, several png images and a textfile which together form the so-called "QA2 report" for the execution block.

The png files can be viewed with standard system tools such as "display" (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) Target visibility amplitude vs. frequency for each spectral window and polarisation
10) Flux calibration model and data (visibility amplitude vs. UV distance)
11) Target visibility amplitude vs. UV distance for each spectral window and polarisation
12) Phase calibrator amplitude and phase vs. frequency for each spectral window and polarisation
13) Target field UV coverage
14) Test image of the target
15) Target imaging synthesized beam (PSF)

The contents of the uid*textfile.txt file is mostly selfexplanatory. 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 (since it might have used additional procedures in imaging). It is mentioned in the README file.

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 (CASA 4.2 and later)*. See the CASA documentation for more details.

If additional diagnostic plots on the visibility data itself are required, the calibrated MeasurementSet (MS) first has to be generated from the raw data (see the next section). Once this is done, the CASA task *plotms* should 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. Starting in the middle of Cycle 1, the products are separated from the raw visibility data (as described in the introduction), and 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 accept the observatory calibration of the data, he/she needs 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 only download the raw data and use the calibration scripts as a guidance and 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. In later stages of the project this will always work via the ALMA Archive Request Handler. See http://almascience.eso.org/alma-data .

5.1 Running the scriptForPI.py

Once the data products tar file is downloaded and unpacked, the user finds a directory structure on disk which is described in section 3 and also at the end of the README file included in the delivery. The following directory tree shows an example:

```
|-- 2012.1.12345.S/
|   |-- sg_ouss_id/
|   |   |-- group_ouss_id/
|   |   |   |-- member_ouss_id/
|   |   |   |   |-- README
|   |   |   |   |-- script/
|   |   |   |   |   |-- scriptForPI.py
|   |   |   |   |   |-- scriptForImaging.py
|   |   |   |   |   |-- scriptForFluxCalibration.py
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.ms.scriptForCalibration.py
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.ms.scriptForCalibration.py
|   |   |   |   |-- calibration/
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.calibration.tgz
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.calibration.plots/
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.calibration.tgz
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.calibration.plots/
|   |   |   |   |-- qa/
|   |   |   |   |   |-- uid___A002_X12345_Xa2d__textfile.txt
|   |   |   |   |   |-- uid___A002_X12345_Xa2d__qa2_part3.png
|   |   |   |   |   |-- uid___A002_X12345_Xa2d__qa2_part2.png
|   |   |   |   |   |-- uid___A002_X12345_Xa2d__qa2_part1.png
|   |   |   |   |   |-- uid___A002_X12345_Xbcd__textfile.txt
|   |   |   |   |   |-- uid___A002_X12345_Xbcd__qa2_part3.png
|   |   |   |   |   |-- uid___A002_X12345_Xbcd__qa2_part2.png
|   |   |   |   |   |-- uid___A002_X12345_Xbcd__qa2_part1.png
|   |   |   |   |-- product/
|   |   |   |   |   |-- NGC1234.cont.dirty.fits
|   |   |   |   |   |-- NGC1234.CO_21.dirty.fits
|   |   |   |   |-- log/
|   |   |   |   |   |-- combined.log
|   |   |   |   |   |-- uid___A002_X12345_Xa2d .calibration.log
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.calibration.log
```

This dataset has two Execution Blocks (EBs) with UIDs “uid___A002_X12345_Xa2d” and “uid___A002_X12345_Xbcd”. In the directory “script” is has therefore two scripts for calibration:

```
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.ms.scriptForCalibration.py
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.ms.scriptForCalibration.py
```

and, since apparently a common flux calibration was necessary, a script to combine the data from the two EBs is present:

```
|   |   |   |   |   |-- scriptForFluxCalibration.py
```

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

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

Unpacking the raw data after download should be done at the top level (in the directory containing the directory 2012.1.12345.S in this example). It should result in an additional directory “raw” at the level of the directory “script”:

```
|   |   |   |   |-- raw/
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.asdm.sdm
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.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
> casapy -c "execfile('scriptForPI.py')"
```

This 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 an alignment of the flux calibrations was deemed necessary by the QA2 process, the scriptForFluxCalibration will be present and will be run as well.

The results will be placed in a new directory “calibrated” at the same level as the directories “raw” and “script”:

```
|   |   |   |   |-- calibrated
|   |   |   |   |   |-- calibrated.ms
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.calibration
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.ms.split.cal
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.calibration
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.ms.split.cal
```

The final calibrated visibility data combined for all EBs can in this case be found in the MS “calibrated.ms”. The intermediate output of CASA to achieve the calibration of each EB is stored in the calibration working directories:

```
|   |   |   |   |   |-- uid___A002_X12345_Xa2d.calibration
|   |   |   |   |   |-- uid___A002_X12345_Xbcd.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).

If the user would like to save disk space and only work with the calibrated data, he/she can either delete all calibration directories (`cd calibrated; rm -rf *.calibration`) or only delete some or all of the intermediate MSs contained in the calibration directories (e.g., `cd calibrated; rm -rf *.calibration/*.ms; rm -rf *.calibration/*.ms.split`).

Starting in mid Cycle 1, the scriptForPI.py will also offer the “SPACESAVING” option to limit the disk space usage during and after the run of the scriptForPI.py. 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
> casapy -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)

With SPACESAVING=0, the required additional diskspace is up to 14 times as large as the delivered data (products and rawdata) 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.

6 Imaging the calibrated visibilities

Once the calibrated visibilities were generated either as described in the previous section or through a customized calibration procedure, the user can proceed to image them.

If lines have been detected over a significant continuum, it should be considered to perform a continuum subtraction using the CASA task *uvcontsub* beforehand. In any case, the imaging script “scriptForImaging.py” from the directory “script” of the delivery can be taken as an example to build on.

The main tool for imaging in CASA is the task *clean*. A detailed description of the usage of this task and its many parameters goes beyond the scope of this document. We refer again to the CASA documentation.

7 Available CASA documentation and help

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

http://casa.nrao.edu/

in particular the user reference and cookbook

http://casa.nrao.edu/Doc/Cookbook/casa_cookbook.pdf

and the ALMA guides wiki

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

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

https://help.almascience.org/

ALMA

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of Europe, North America and East Asia in cooperation with the Republic of Chile. ALMA is funded in Europe by the European Organization for Astronomical Research in the Southern Hemisphere (ESO), in North America by the U.S. National Science Foundation (NSF) in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and in East Asia by the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Academia Sinica (AS) in Taiwan. ALMA construction and operations are led on behalf of Europe by ESO, on behalf of North America by the National Radio Astronomy Observatory (NRAO), which is managed by Associated Universities, Inc. (AUI) and on behalf of East Asia by the National Astronomical Observatory of Japan (NAOJ). The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.

ESO NAOJ National Astronomical Observatory of Japan NRAO