

---

**ALMAQSO**

**Akimasa Nishida, Yuki Yoshimura**

**Dec 29, 2025**



**CONTENTS:**

<b>1</b>	<b>almaqso package</b>	<b>1</b>
1.1	Submodules . . . . .	1
1.2	almaqso.almaqso module . . . . .	1
1.3	Module contents . . . . .	2
<b>2</b>	<b>Architecture</b>	<b>3</b>
2.1	System context . . . . .	3
2.2	Container . . . . .	4
2.3	Component . . . . .	4
	<b>Python Module Index</b>	<b>7</b>



## ALMAQSO PACKAGE

## 1.1 Submodules

## 1.2 almaqso.almaqso module

**class** `Almaqso`(*target: list[str]*, *band: int*, *cycle: str = ''*, *work\_dir: str = './'*, *casapath: str = 'casa'*)

Bases: `object`

**process**(*n\_parallel: int = 1*, *skip\_previous\_successful: bool = False*, *do\_tclean: bool = False*, *tclean\_mode: list[str] = ['mfs']*, *tclean\_weightings: tuple[str, str] = ('natural', '')*, *do\_selfcal: bool = False*, *kw\_selfcal: dict[str, object] = {}*, *do\_export\_fits: bool = False*, *remove\_casa\_images: bool = False*, *remove\_asdm: bool = False*, *remove\_intermediate: bool = False*) → `None`

Download and process ALMA data.

**Parameters**

- **n\_parallel** (*int*) – The number of the parallel execution.
- **do\_tclean** (*bool*) – Perform tclean. Default is False.
- **skip\_previous\_successful** (*bool*) – Skip processing for previously successful tasks. Default is False.
- **tclean\_mode** (*list[str]*) – List of imaging specmodes for tclean. “mfs” creates a MFS image, “mfs\_spw” creates MFS images for each spw, and “cube” creates a cube image. Default is [“mfs”].
- **tclean\_weightings** (*tuple[str, str]*) – Weighting scheme and robust parameter for tclean. Second element is the robust parameter for briggs weighting. Default is (“natural”, “”).
- **do\_selfcal** (*bool*) – Perform self-calibration. Default is False.
- **kw\_selfcal** (*dict[str, object]*) – Parameters for the self-calibration and *tclean* task.
- **do\_export\_fits** (*bool*) – Export the final image to FITS format. Default is False.
- **remove\_casa\_images** (*bool*) – Remove the CASA images after processing. This option only works if *do\_tclean* is True. Default is False.
- **remove\_asdm** (*bool*) – Remove the ASDM files after processing. Default is False.
- **remove\_intermediate** (*bool*) – Remove the intermediate files after processing. Log of CASA will be retained. Default is False.

**Returns**

`None`

**analysis()** → None

Perform the analysis.

## 1.3 Module contents

**class Almqso**(*target: list[str], band: int, cycle: str = "", work\_dir: str = './', casapath: str = 'casa'*)

Bases: object

**analysis()** → None

Perform the analysis.

**process**(*n\_parallel: int = 1, skip\_previous\_successful: bool = False, do\_tclean: bool = False, tclean\_mode: list[str] = ['mfs'], tclean\_weightings: tuple[str, str] = ('natural', ''), do\_selfcal: bool = False, kw\_selfcal: dict[str, object] = {}, do\_export\_fits: bool = False, remove\_casa\_images: bool = False, remove\_asdm: bool = False, remove\_intermediate: bool = False*) → None

Download and process ALMA data.

### Parameters

- **n\_parallel** (*int*) – The number of the parallel execution.
- **do\_tclean** (*bool*) – Perform tclean. Default is False.
- **skip\_previous\_successful** (*bool*) – Skip processing for previously successful tasks. Default is False.
- **tclean\_mode** (*list[str]*) – List of imaging specmodes for tclean. “mfs” creates a MFS image, “mfs\_spw” creates MFS images for each spw, and “cube” creates a cube image. Default is [“mfs”].
- **tclean\_weightings** (*tuple[str, str]*) – Weighting scheme and robust parameter for tclean. Second element is the robust parameter for briggs weighting. Default is (“natural”, “”).
- **do\_selfcal** (*bool*) – Perform self-calibration. Default is False.
- **kw\_selfcal** (*dict[str, object]*) – Parameters for the self-calibration and *tclean* task.
- **do\_export\_fits** (*bool*) – Export the final image to FITS format. Default is False.
- **remove\_casa\_images** (*bool*) – Remove the CASA images after processing. This option only works if *do\_tclean* is True. Default is False.
- **remove\_asdm** (*bool*) – Remove the ASDM files after processing. Default is False.
- **remove\_intermediate** (*bool*) – Remove the intermediate files after processing. Log of CASA will be retained. Default is False.

### Returns

None

## ARCHITECTURE

### 2.1 System context

This section provides a general overview of the ALMAQSO system within user (researcher) and external system interactions.

#### 2.1.1 Purpose

This diagram explains how ALMAQSO fits into the surrounding ecosystem and how researchers interact with external systems.

#### 2.1.2 Audience

- Researchers using ALMAQSO
- New contributors
- Developers unfamiliar with CASA

#### 2.1.3 Components

- **User:** Users who conduct data analysis using ALMAQSO. They are typically researchers in astronomy.
- **ALMAQSO:** This software system that provides automated data retrieval and analysis for ALMA calibration sources.
- **External Systems:** CASA, analysisUtils, ALMA Science Archive and local file systems that interact with ALMAQSO.

#### 2.1.4 C4-Context Diagram

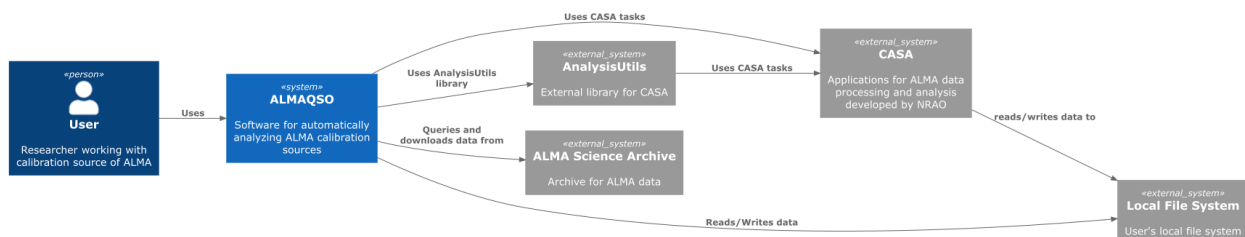


Fig. 1: C4-Context Diagram. (Web version: Click to show full-size diagram.)

## 2.2 Container

This section provides an overview of ALMAQSO showing the boundary of container (executable, library, database and so on) within the system. However, ALMAQSO is a small project and consists of only one container, ALMAQSO Library.

### 2.2.1 Purpose

This diagram explains the high-level technology choices made within ALMAQSO and how responsibilities are distributed across the containers.

### 2.2.2 Audience

- Developers working on ALMAQSO
- New contributors

### 2.2.3 Components

Here, we explain only new components that were not explained in the System Context diagram.

- **ALMAQSO Library:** A Python library that provides functionalities for downloading, processing, and analyzing ALMA calibration source data. Python is widely used in the radio observational astronomy community, making it a suitable choice for this library.

### 2.2.4 C4-Container Diagram

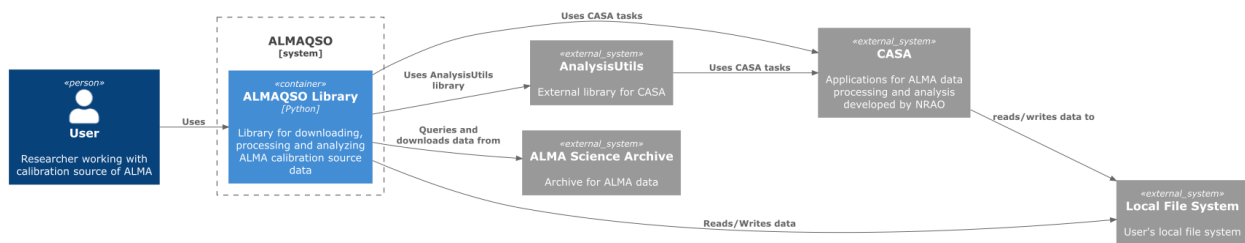


Fig. 2: C4-Container Diagram. (Web version: [Click to show full-size diagram.](#))

## 2.3 Component

This section provides a detailed overview of the internal structure of the ALMAQSO Library container, breaking it down into its modules and classes. This section may be updated frequently since the development phase is moving from PoC (Proof of Concept) to a well-structured library.

### 2.3.1 Purpose

This diagram explains how the ALMAQSO Library is structured internally, showing the main components and their interactions.

### 2.3.2 Audience

- Developers working on ALMAQSO
- New contributors



### 2.3.3 Components

Here, we explain the main components of the ALMAQSO Library:

- **Almaqso Class:** The main class that provides the public API for users to interact with the library.
- **Logger Manager:** A component responsible for initializing and configuring logging within the library.
- **Query Class:** A class that handles querying the ALMA Science Archive for calibration source data.
- **Download Function:** A function that manages the downloading of data from the ALMA Science Archive.
- **Process Class:** A class that encapsulates the logic for processing downloaded data using CASA tasks.
- **Analysis Functions or Classes:** Components that provide functionalities for analyzing the processed data. This is not yet implemented.

### 2.3.4 C4-Component Diagram

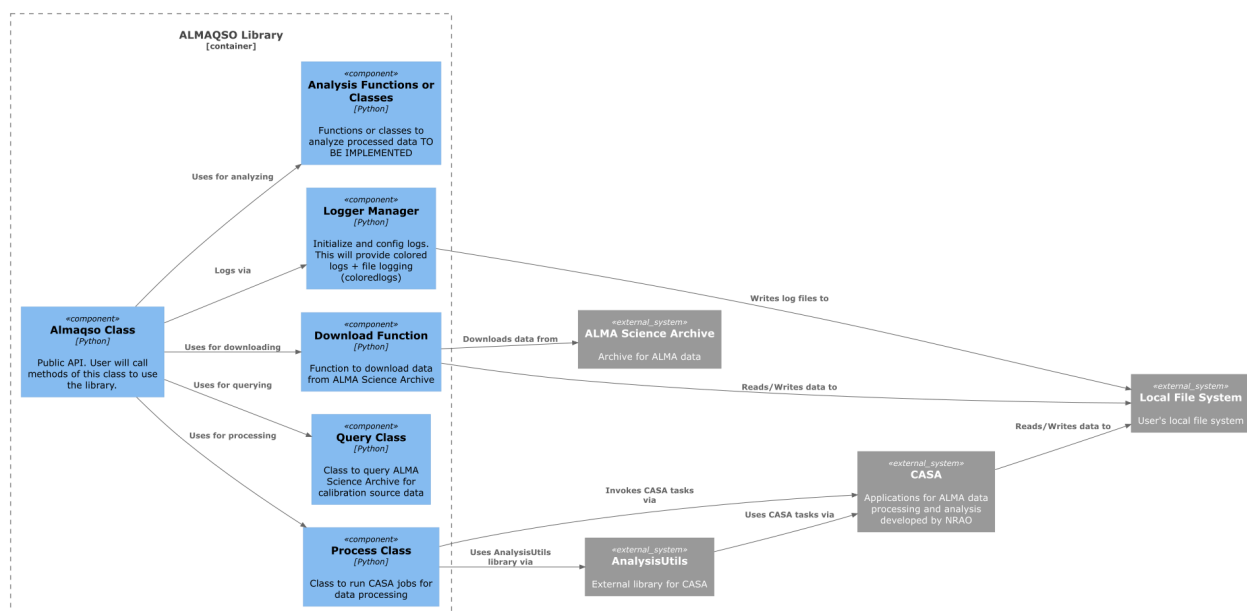


Fig. 3: C4-Component Diagram. (Web version: Click to show full-size diagram.)



## PYTHON MODULE INDEX

### a

almaqso, [2](#)

almaqso.almaqso, [1](#)

## INDEX

- \spxentryalmaqso
  - \spxentrymodule, [2](#)
- \spxentryAlmaqso\spxextraclass in almaqso, [2](#)
- \spxentryAlmaqso\spxextraclass in almaqso.almaqso, [1](#)
- \spxentryalmaqso.almaqso
  - \spxentrymodule, [1](#)
- \spxentryanalysis()\spxextraAlmaqso method, [2](#)
- \spxentrymodule
  - \spxentryalmaqso, [2](#)
  - \spxentryalmaqso.almaqso, [1](#)
- \spxentryprocess()\spxextraAlmaqso method, [1](#), [2](#)