

Introduction to



EUROPEAN ARC
ALMA Regional Centre || UK

Ana Karla Díaz-Rodríguez

MANCHESTER
1824

What is CASA?

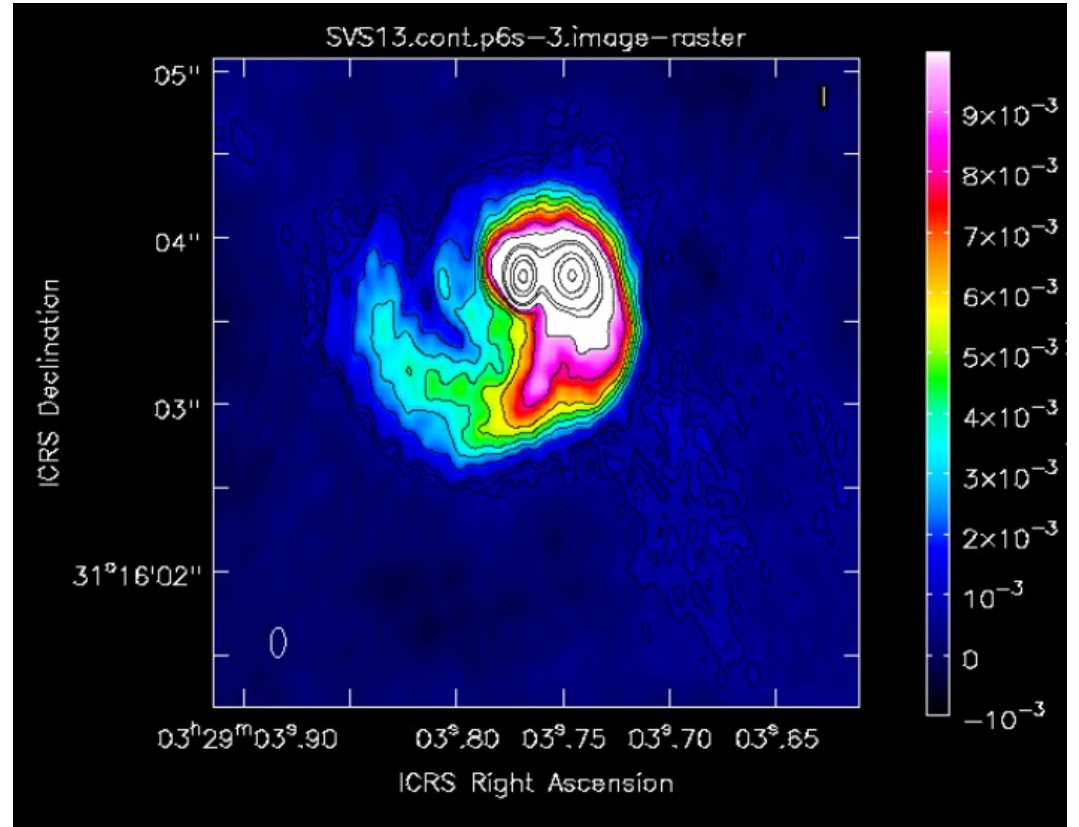
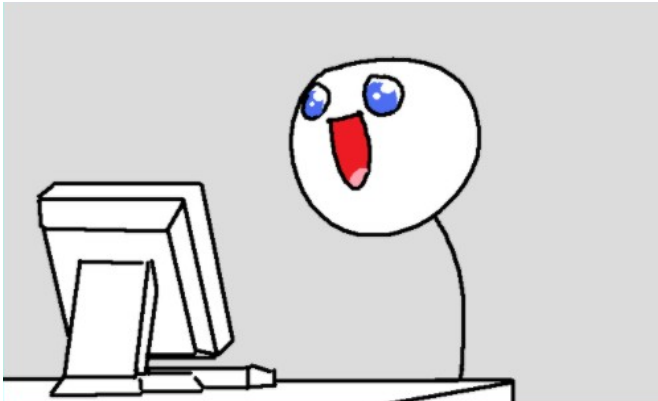
- **C**ommon **A**stronomy **S**oftware **A**pplication
- Used for calibration, imaging and analysis of interferometric data
- Runs on C++ and has a iPython interface
- Developed for Linux (RedHat) and Mac
- Current version is CASA 6.4. We will use CASA 6.2 (pipeline)

Get CASA [here](#)

ALMA



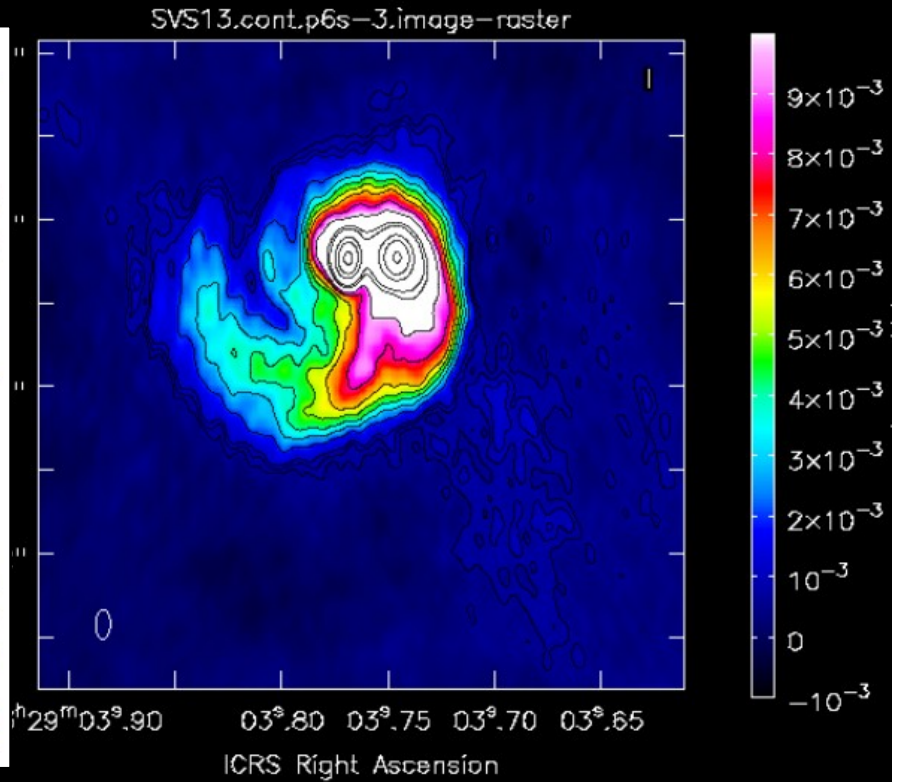
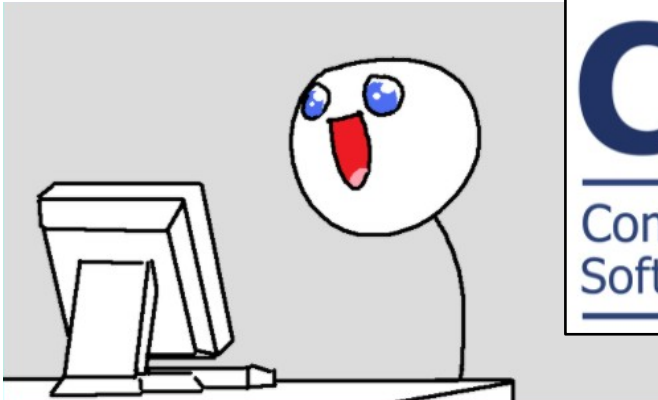
Happy radioastronomer

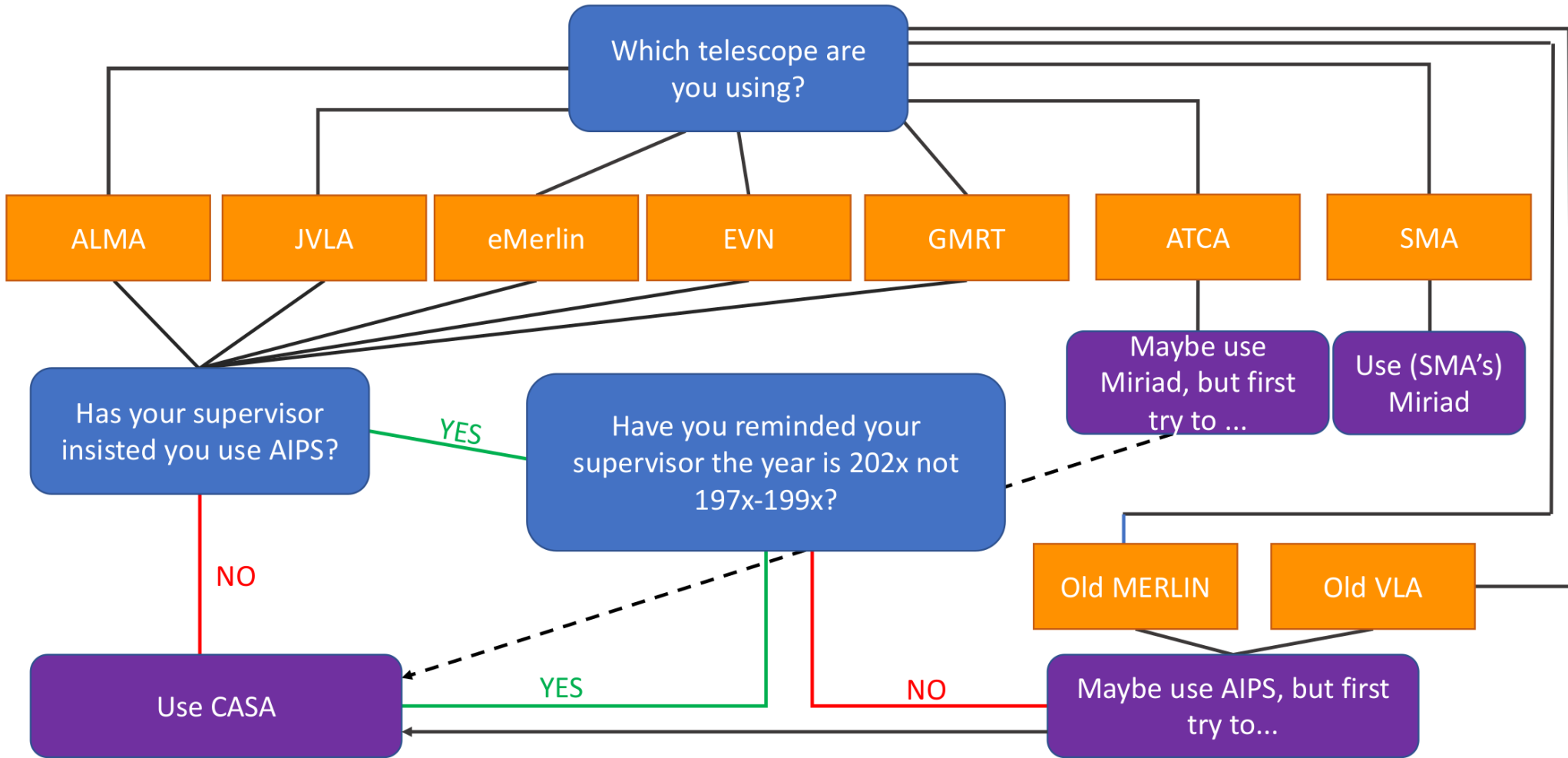


ALMA



Happy radioastronomer





(borrowed from Adam Avison)

Starting CASA

- Type 'casa' in a terminal

Will start up the iPython interface in the terminal and launch the Logger GUI

```
anika@Enterprise: ~$ casa
optional configuration file config.py not found, continuing CASA startup without it
Using user-supplied startup.py at /home/anika/.casa/startup.py
IPython 7.15.0 -- An enhanced Interactive Python.

Using matplotlib backend: agg
Telemetry initialized. Telemetry will send anonymized usage statistics to NRAO.
You can disable telemetry by adding the following line to the config.py file in your
rcdir (e.g. ~/.casa/config.py):
telemetry_enabled = False
--> CrashReporter initialized.
imported casatasks and casatools individually
/home/anika/Documents/CASA_releases/casa-6.2.1-7-pipeline-2021.2.0.128/lib/python3.6/site-packages/pyfits/_init_.py:22: PyFITSDeprecationWarning: PyFITS is deprecated, please use astropy.io.fits
  PyFITSDeprecationWarning) # noqa
CASA 6.2.1.7 -- Common Astronomy Software Applications [6.2.1.7]

CASA <1>
```

```
Log Messages (/home/anika/casa-20220208-165017.log)
message:
Filter: Time
Origin Message
::casa
::casa
::casa
2022-02-08 16:50:19 INFO ::casa
2022-02-08 16:50:19 INFO ::casa CASA Version PIPELINE 6.2.1.7
2022-02-08 16:50:19 INFO ::casa
2022-02-08 16:50:19 INFO ::casa Found an existing telemetry logfile: /home/anika/.casa/casastats-6.2.1.7-160222129cd1c7179-20220208-092809.log
2022-02-08 16:50:19 INFO ::casa Telemetry log file: /home/anika/.casa/casastats-6.2.1.7-160222129cd1c7179-20220208-092809.log
2022-02-08 16:50:19 INFO ::casa Checking telemetry submission interval
2022-02-08 16:50:19 INFO ::casa Telemetry submit interval not reached. Not submitting data.
2022-02-08 16:50:19 INFO ::casa Next telemetry data submission in: 6 days, 16:37:50.256760
2022-02-08 16:50:20 INFO ::casa
2022-02-08 16:50:20 INFO ::casa
2022-02-08 16:50:20 INFO ::casa Checking Measures tables in data repository sub-directory /home/anika/Documents/CASA_releases/casa-6.2.1-7-pipeline-
2022-02-08 16:50:20 INFO ::casa IERSep2000 (version date, last date in table (UTC)): 2022/01/19/15:00, 2021/12/19/00:00:00
2022-02-08 16:50:20 INFO ::casa IERSep97 (version date, last date in table (UTC)): 2022/01/19/15:00, 2021/12/19/00:00:00
2022-02-08 16:50:20 INFO ::casa IERSpredict (version date, last date in table (UTC)): 2022/01/23/15:00, 2022/04/23/00:00:00
2022-02-08 16:50:20 INFO ::casa TAI_UTC (version date, last date in table (UTC)): 2022/01/11/15:00, 2017/01/01/00:00:00
Insert Message: + Lock scroll
```

CASA Tasks and Tools

TASKS:

Front end, user friend command line functions for data reduction, manipulation in CASA.

Built upon the TOOL kit functions available in CASA.

Typically have a bit more functionality than functions available in GUIs e.g. viewer

Tasks list

VS

TOOLS:

'Under the hood' basic functions upon which tasks are built.

Perform simple tasks but can be useful in image manipulation and some simulation tasks.

Tools list

CASA Tasks

- Interface to the toolkit
- Perform a single job such as loading, plotting, flagging, etc
- Contain a set of user definable parameters
- E.g.: **importfits**(fitsimage='my_image.fits',
 imagenamename='my_image.im')

Getting help in CASA

- Type 'inp importfits'

```
IPython: home/anika
CASA <1>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''          # Name of input image FITS file
imagename      = ''          # Name of output CASA image
whichrep       = 0           # If fits image has multiple coordinate
                           # reps, choose one.
whichhdu       = -1          # If fits file contains multiple
                           # images, choose one (0 = first HDU, -1
                           # = first valid image).
zeroblanks     = True        # Set blanked pixels to zero (not NaN)
overwrite      = False       # Overwrite output file if it exists?
defaultaxes    = False       # Add the default 4D coordinate axes
                           # where they are missing; value True
                           # requires setting defaultaxesvalues
defaultaxesvalues = []       # List of values to assign to added
                           # degenerate axes when
                           # defaultaxes==True
```

inp <<task_name>>

Will give you info about the parameters of a specific task

(some tasks to try: plotms, tclean, imfit, imhead)

Getting help in CASA

- Type 'help importfits'

```
IPython: home/anika
Help on _importfits in module casashell.private.importfits object:

class _importfits(builtins.object)
| importfits ---- Convert an image FITS file into a CASA image
|
| Convert an image FITS file into a CASA image
|
| ----- parameter descriptions -----
|
| fitsimage          Name of input image FITS file
|                    Default: none
|
|                    Example: fitsimage='3C273XC1.fits'
|
| imagename          Name of output CASA image
|                    Default: none
|
|                    Example: fitsimage='3C273XC1.image'
|
|
```

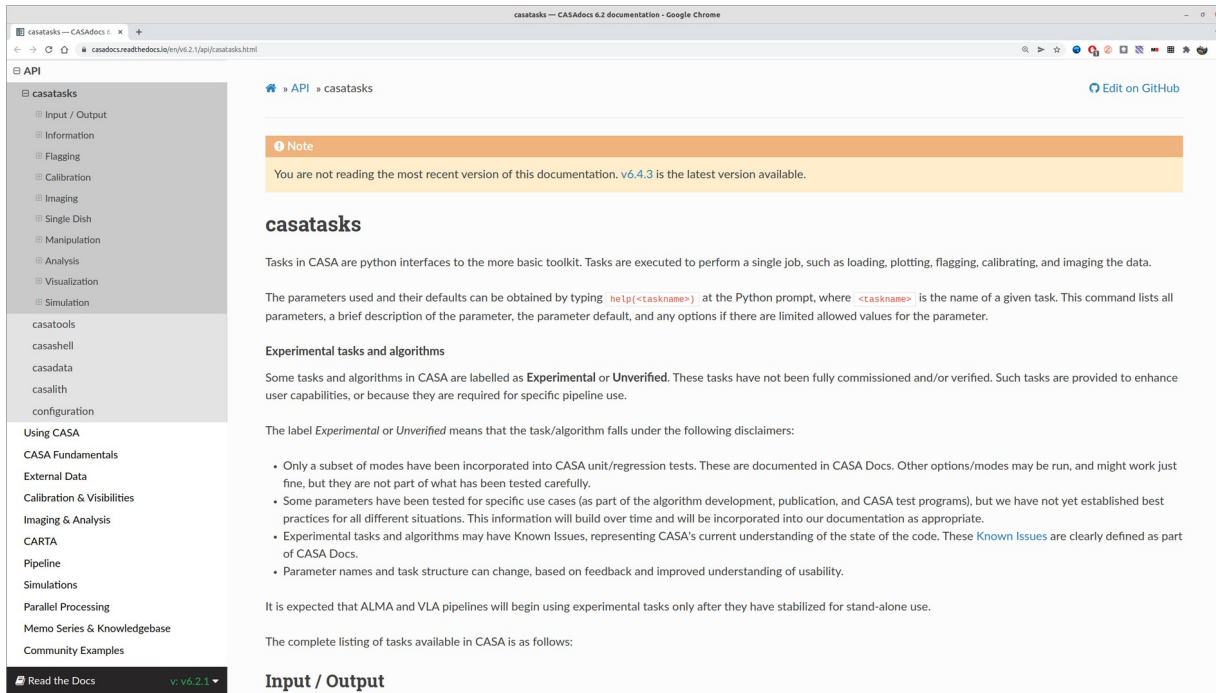
help <<task_name>>

Will give you detailed info about the task

Exit with 'q'

Getting help in CASA

- Type ‘doc plotms’



The screenshot shows the CASA documentation website in a Google Chrome browser. The address bar shows the URL `casadocs.readthedocs.io/en/v6.2.1/html/casatasks.html`. The page title is "casatasks — CASAdocs 6.2 documentation". A navigation sidebar on the left lists various categories like "casatasks", "casatools", "Using CASA", etc. The main content area features a "Note" box with a warning icon, stating: "You are not reading the most recent version of this documentation. v6.4.3 is the latest version available." Below this, the "casatasks" section is introduced, followed by a paragraph explaining that tasks are Python interfaces to the basic toolkit. A sub-section titled "Experimental tasks and algorithms" explains that some tasks are labeled as "Experimental" or "Unverified". A list of disclaimers follows, detailing the limitations of experimental tasks. The page concludes with the text "The complete listing of tasks available in CASA is as follows:" and a heading for "Input / Output".

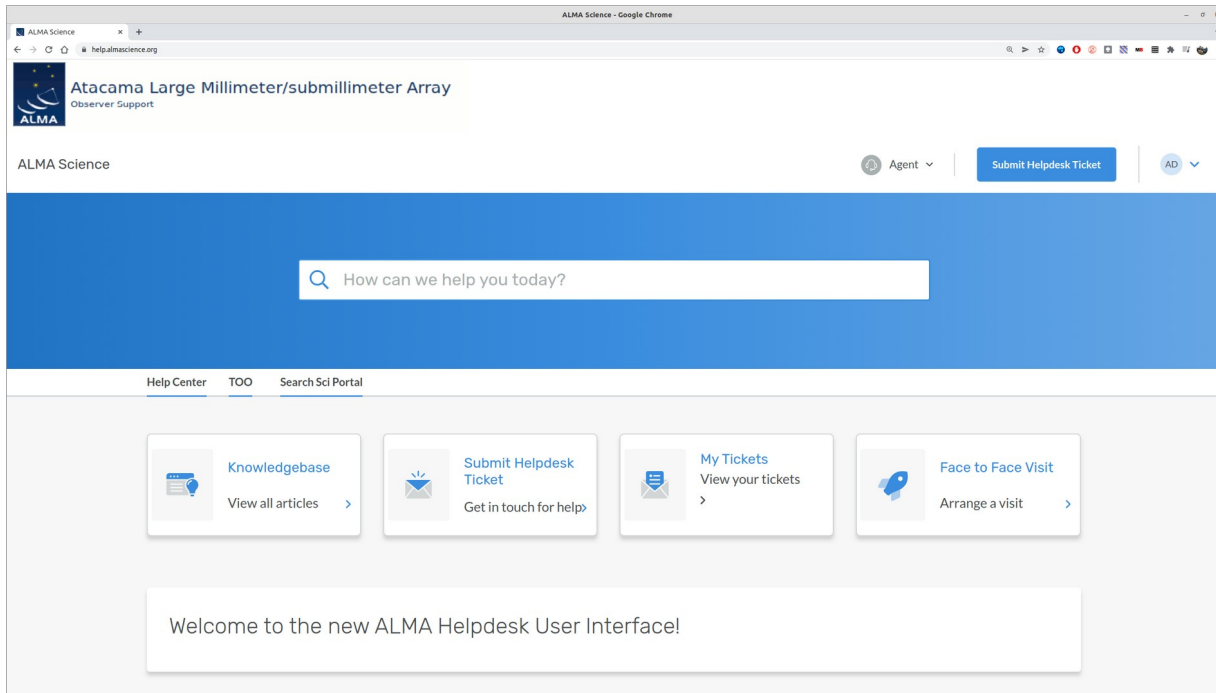
doc <<task_name>>

Will take you to the online documentation

Press ‘enter’ to continue working in the terminal

Getting help in CASA

- ALMA Helpdesk



Submit your ticket to the **Data Reduction Department** and state your **CASA version** and **operating system**

Working with CASA

To execute a task:

1. `default <<task_name>>`, to set the parameters to the default value
2. `inp <<task_name>>`, to check the parameters
3. `par1 = a, ...`, to fill in the parameters you need
4. `inp <<task_name>>`, to check the values
5. `<<task_name>>`, to run the task

Example

- `cd` to the folder 'material' and `ls` there to see what we have

```
CASA <2>: ls  
12.fits* 27.fits* 3.fits* 40.fits* casa-20220210-093719.log casa-20220210-094403.log
```

(Simple shell commands like `cd` and `ls` work inside CASA; more complicated ones work prefixed by `!`, e.g. `!cp`)

Example

- `cd` to the folder 'material' and `ls` there to see what we have

```
CASA <2>: ls  
12.fits* 27.fits* 3.fits* 40.fits* casa-20220210-093719.log casa-20220210-094403.log
```

- Convert **12.fits** into a casa image with `importfits`

Example

- `cd` to the folder 'material' and `ls` there to see what we have

```
CASA <2>: ls  
12.fits* 27.fits* 3.fits* 40.fits* casa-20220210-093719.log casa-20220210-094403.log
```

- Convert **12.fits** into a casa image with `importfits`

```
1 CASA <3>: default importfits  
-----> default(importfits)  
-----> default(importfits)
```


Example

- `cd` to the folder 'material' and `ls` there to see what we have

```
CASA <2>: ls  
12.fits* 27.fits* 3.fits* 40.fits* casa-20220210-093719.log casa-20220210-094403.log
```

- Convert **12.fits** into a casa image with `importfits`

1

```
CASA <3>: default importfits  
-----> default(importfits)  
-----> default(importfits)
```

2

```
CASA <4>: inp importfits  
-----> inp(importfits)  
-----> inp(importfits)  
# importfits -- Convert an image FITS file into a CASA image  
fitsimage      = ''          # Name of input image FITS file  
imagenname     = ''          # Name of output CASA image  
whichrep       = 0           # If fits image has multiple coordinate reps,  
whichhdu       = -1          # If fits file contains multiple images, cho  
# = first valid image).  
zeroblanks     = True        # Set blanked pixels to zero (not NaN)
```

Example

- `cd` to the folder 'material' and `ls` there to see what we have

```
CASA <2>: ls
12.fits* 27.fits* 3.fits* 40.fits* casa-20220210-093719.log casa-20220210-094403.log
```

- Convert **12.fits** into a casa image with `importfits`

```
1 CASA <3>: default importfits
-----> default(importfits)
-----> default(importfits)
```

```
2 CASA <4>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''          # Name of input image FITS file
imagenname     = ''          # Name of output CASA image
whicnrep       = 0           # If fits image has multiple coordinate reps.
whichhdu       = -1         # If fits file contains multiple images, choo
# = first valid image).
zeroblanks     = True       # Set blanked pixels to zero (not NaN)
```

```
3 CASA <5>: fitsimage='12.fits'
CASA <6>: imagenname='12.image'
```

Example

- Convert **12.fits** into a casa image with **importfits**

1

```
CASA <3>: default importfits
-----> default(importfits)
-----> default(importfits)
```

2

```
CASA <4>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''           # Name of input image FITS file
imagenname     = ''           # Name of output CASA image
whicnrep       = 0            # If fits image has multiple coordinate reps,
whichhdu       = -1           # If fits file contains multiple images, choo
# = first valid image).
zeroblanks     = True         # Set blanked pixels to zero (not NaN)
```

3

```
CASA <5>: fitsimage='12.fits'
CASA <6>: imagenname='12.image'
```

4

```
CASA <7>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = '12.fits'    # Name of input image FITS file
imagenname     = '12.image'   # Name of output CASA image
whicnrep       = 0            # If fits image has multiple coordinate reps, choose one.
whichhdu       = -1           # If fits file contains multiple images, choose one (0 = first HDU, -1
# = first valid image).
```

Example

- Convert **12.fits** into a casa image with **importfits**

```
1 CASA <3>: default importfits
-----> default(importfits)
-----> default(importfits)
```

```
2 CASA <4>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''                # Name of input image FITS file
imagenam       = ''                # Name of output CASA image
whicnrep       = 0                 # If fits image has multiple coordinate reps.
whichhdu       = -1               # If fits file contains multiple images, choo
# = first valid image).
zeroblanks     = True             # Set blanked pixels to zero (not NaN)
```

```
3 CASA <5>: fitsimage='12.fits'
CASA <6>: imagename='12.image'
```

```
4 CASA <7>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = '12.fits'         # Name of input image FITS file
imagenam       = '12.image'       # Name of output CASA image
whicnrep       = 0                 # If fits image has multiple coordinate reps, choose one.
whichhdu       = -1               # If fits file contains multiple images, choose one (0 = first HDU, -1
# = first valid image).
```

```
5 CASA <8>: importfits
-----> importfits()
-----> importfits()
```

Example

- **ls** again in the folder **'material'** to see what we have

```
CASA <9>: ls
12.fits*  27.fits*  40.fits*  casa-20220210-094403.log
12.image/ 3.fits*  casa-20220210-093719.log  importfits.last
```

Example

- `ls` again in the folder 'material' to see what we have

```
CASA <9>: ls
12.fits*  27.fits*  40.fits*  casa-20220210-094403.log
12.image/ 3.fits*  casa-20220210-093719.log  importfits.last
```

word of caution: the parameters were set **globally**

Example

- `ls` again in the folder **'material'** to see what we have

```
CASA <9>: ls
12.fits*  27.fits*  40.fits*  casa-20220210-094403.log
12.image/ 3.fits*  casa-20220210-093719.log  importfits.last
```

word of caution: the parameters were set **globally**

- Check parameter values of `immath`

Example

- **ls** again in the folder **'material'** to see what we have

```
CASA <9>: ls
12.fits* 27.fits* 40.fits* casa-20220210-094403.log
12.image/ 3.fits* casa-20220210-093719.log importfits.last
```

word of caution: the parameters were set **globally**

- Check parameter values of **immath**

```
-----> inp(immath)
# immath -- Perform math operations on images
image_name = '12.image' # a list of input images
mode = 'evalexpr' # mode for math operation (evalexpr, spix, pola, poli, lpo
outfile = 'immath_results.im' # File where the output is saved
expr = '' # Mathematical expression using images
varnames = '' # a list of variable names to use with the image files
mask = '' # Mask to use. Default is none.
region = '' # Region selection. Default is to use the full image.
help = '' # Display this help message. Default is to not display.
```


Working with CASA

To execute a task (recommended):

1. '`default <<task_name>>`', to set the parameters to the default value
2. '`inp <<task_name>>`', to check the parameters
3. '`<<task_name>>(par1 = a, ...)`', to fill in the parameters you need and run the task

Example

- Convert **27.fits** into a casa image with **importfits**

Example

- Convert **27.fits** into a casa image with **importfits**

1

```
CASA <3>: default importfits
-----> default(importfits)
-----> default(importfits)
```

2

```
CASA <4>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''          # Name of input image FITS file
imagenname     = ''          # Name of output CASA image
whichrep       = 0           # If fits image has multiple coordinate reps.
whichhdu       = -1          # If fits file contains multiple images, choo
# = first valid image).
zeroblanks     = True        # Set blanked pixels to zero (not NaN)
```

Example

- Convert **27.fits** into a casa image with **importfits**

1

```
CASA <3>: default importfits
-----> default(importfits)
-----> default(importfits)
```

2

```
CASA <4>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''          # Name of input image FITS file
imagenname     = ''          # Name of output CASA image
whichrep       = 0           # If fits image has multiple coordinate reps.
whichhdu       = -1         # If fits file contains multiple images, choo
# = first valid image).
zeroblanks     = True       # Set blanked pixels to zero (not NaN)
```

3

```
CASA <14>: importfits(fitsimage='27.fits',imagenname='27.image')
```

Example

- Convert **27.fits** into a casa image with **importfits**

```
1 CASA <3>: default importfits
-----> default(importfits)
-----> default(importfits)
```

```
2 CASA <4>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''                # Name of input image FITS file
imagenname     = ''                # Name of output CASA image
whichrep       = 0                 # If fits image has multiple coordinate reps.
whichhdu       = -1                # If fits file contains multiple images, cho
# = first valid image).
zeroblanks     = True              # Set blanked pixels to zero (not NaN)
```

```
3 CASA <14>: importfits(fitsimage='27.fits',imagenname='27.image')
```

- **ls** again in the folder **'material'** to see what we have

```
CASA <15>: ls
12.fits*    27.fits*    3.fits*    casa-20220210-093719.log  importfits.last
12.image/  27.image/   40.fits*   casa-20220210-094403.log
```

Example

- Convert **27.fits** into a casa image with **importfits**

```
1 CASA <3>: default importfits
-----> default(importfits)
-----> default(importfits)
```

```
2 CASA <4>: inp importfits
-----> inp(importfits)
-----> inp(importfits)
# importfits -- Convert an image FITS file into a CASA image
fitsimage      = ''                # Name of input image FITS file
imagename      = ''                # Name of output CASA image
whichrep       = 0                 # If fits image has multiple coordinate reps.
whichhdu       = -1                # If fits file contains multiple images, choo
# = first valid image).
zeroblanks     = True              # Set blanked pixels to zero (not NaN)
```

```
3 CASA <14>: importfits(fitsimage='27.fits',imagename='27.image')
```

- **ls** again in the folder **'material'** to see what we have

```
CASA <15>: ls
12.fits*  27.fits*  3.fits*  casa-20220210-093719.log  importfits.last
12.image/ 27.image/ 40.fits*  casa-20220210-094403.log
```

- Check parameter values of **immath** and **importfits**

Keep record of your work

```
default importfits
inp importfits

importfits(fitsimage = '12.fits', imagename = '12.image')
importfits(fitsimage = '27.fits', imagename = '27.image')
```

Write the commands in a text file and paste them into the terminal

```
CASA <40>: ls
12.fits* 27.fits* 3.fits* 40.fits* casa-20220210-093719.log casa-20220210-094403.log importfits.last

CASA <41>: importfits(fitsimage = '12.fits', imagename = '12.image')
...: importfits(fitsimage = '27.fits', imagename = '27.image')
...:

CASA <42>: ls
12.fits* 27.fits* 3.fits* casa-20220210-093719.log importfits.last
12.image/ 27.image/ 40.fits* casa-20220210-094403.log

CASA <43>:
```

Keep record of your work

```
fitsimage = ['12.fits', '27.fits']

for fits in fitsimage:
    importfits(fitsimage = fits, imagename = fits[:-5]+'image')
```

Allows you doing things
in a clever way

```
CASA <11>: ls
12.fits*  3.fits*  casa-20220210-093719.log  casa-20220210-120941.log  casa-20220210-121614.log
27.fits*  40.fits*  casa-20220210-094403.log  casa-20220210-121037.log  intro_CASA.py

CASA <12>: fitsimage = ['12.fits', '27.fits']
...:
...: for fits in fitsimage:
...:     importfits(fitsimage = fits, imagename = fits[:-5]+'image')
...:

CASA <13>: ls
12.fits*  27.fits*  3.fits*  casa-20220210-093719.log  casa-20220210-120941.log  casa-20220210-121614.log  intro_CASA.py
12.image/ 27.image/ 40.fits*  casa-20220210-094403.log  casa-20220210-121037.log  importfits.last

CASA <14>:
```


CASA Data

1. Visibilities (uv data)

CASA Data

1. Visibilities (uv data)

Raw: ADMS (ALMA Science Data Model)

→ for data storage

CASA Data

1. Visibilities (uv data)

Raw: ADMS (ALMA Science Data Model)

→ for data storage

Imported: MS (Measurement Set)

→ for data reduction

CASA Data

1. Visibilities (uv data)

```
IPython: Downloads/uid__A002_Xd98580_X354.ms.calibrated
CASA <39>: cd uid__A002_Xd98580_X354.ms.calibrated/
/home/anika/Downloads/uid__A002_Xd98580_X354.ms.calibrated

CASA <40>: ls
ANTENNA/          DATA_DESCRIPTION/ Source.xml          table.f13          table.f19          table.f22_TSM1    table.f3
ASDM_ANTENNA/    DataDescription.xml SPECTRAL_WINDOW/  table.f14          table.f2           table.f22_TSM2    table.f4
ASDM_CALATMOSPHERE/ FEED/             SpectralWindow.xml table.f15          table.f20          table.f23          table.f5
ASDM_CALPOINTING/ FIELD/            STATE/            table.f16          table.f20_TSM0     table.f23_TSM1    table.f6
ASDM_CALWVR/     FLAG_CMD/        SYSCAL/          table.f17          table.f21          table.f23_TSM2    table.f7
ASDM_EXECBLOCK/ HISTORY/          SYSPower/        table.f17_TSM1    table.f21_TSM1     table.f24          table.f8
ASDM_RECEIVER/  OBSERVATION/    table.dat        table.f17_TSM2    table.f21_TSM2     table.f24_TSM1    table.f9
ASDM_SBSUMMARY/ POINTING/        table.f1         table.f17_TSM3    table.f21_TSM3     table.f24_TSM2    table.info
ASDM_SOURCE/    POLARIZATION/   table.f10        table.f17_TSM4    table.f21_TSM4     table.f24_TSM3    table.lock
ASDM_STATION/   PROCESSOR/       table.f11        table.f17_TSM5    table.f21_TSM5     table.f24_TSM4    WEATHER/
CALDEVICE/     SOURCE/          table.f12        table.f18          table.f22          table.f24_TSM5
```

CASA Data

2. Images

```
IPython: CASA/material
CASA <54>: ls
12.fits* 27.fits* 3.fits* 40.fits* casa-20220210-093719.log casa-20220210-121037.log importfits.last
12.image/ 27.image/ 3.image/ 40.image/ casa-20220210-094403.log casa-20220210-121614.log intro_CASA.py
12.txt 27.txt 3.txt 40.txt casa-20220210-120941.log imfit.last

CASA <55>: ls 12.image
logtable/ mask0/ table.dat table.f0 table.f0_TSM0 table.info table.lock

CASA <56>: ls 12.image/logtable/
table.dat table.f0 table.info table.lock

CASA <57>: ls 12.image/mask0/
table.dat table.f0 table.f0_TSM0 table.info table.lock

CASA <58>: imview
-----> imview()
-----> imview()

CASA <59>:
```

Scripts

- You can write **Python* scripts** with CASA commands and execute them

Useful when you have to process several images/sources in a similar way

* Python 3 for CASA 6.x and Python 2 for CASA 5.x and earlier

Scripts

- You can write **Python* scripts** with CASA commands and execute them

inside CASA with

`execfile('my_script.py')`

Useful when you have to process several images/sources in a similar way

* Python 3 for CASA 6.x and Python 2 for CASA 5.x and earlier

Scripts

- You can write **Python* scripts** with CASA commands and execute them

inside CASA with

`execfile('my_script.py')`

or from a terminal

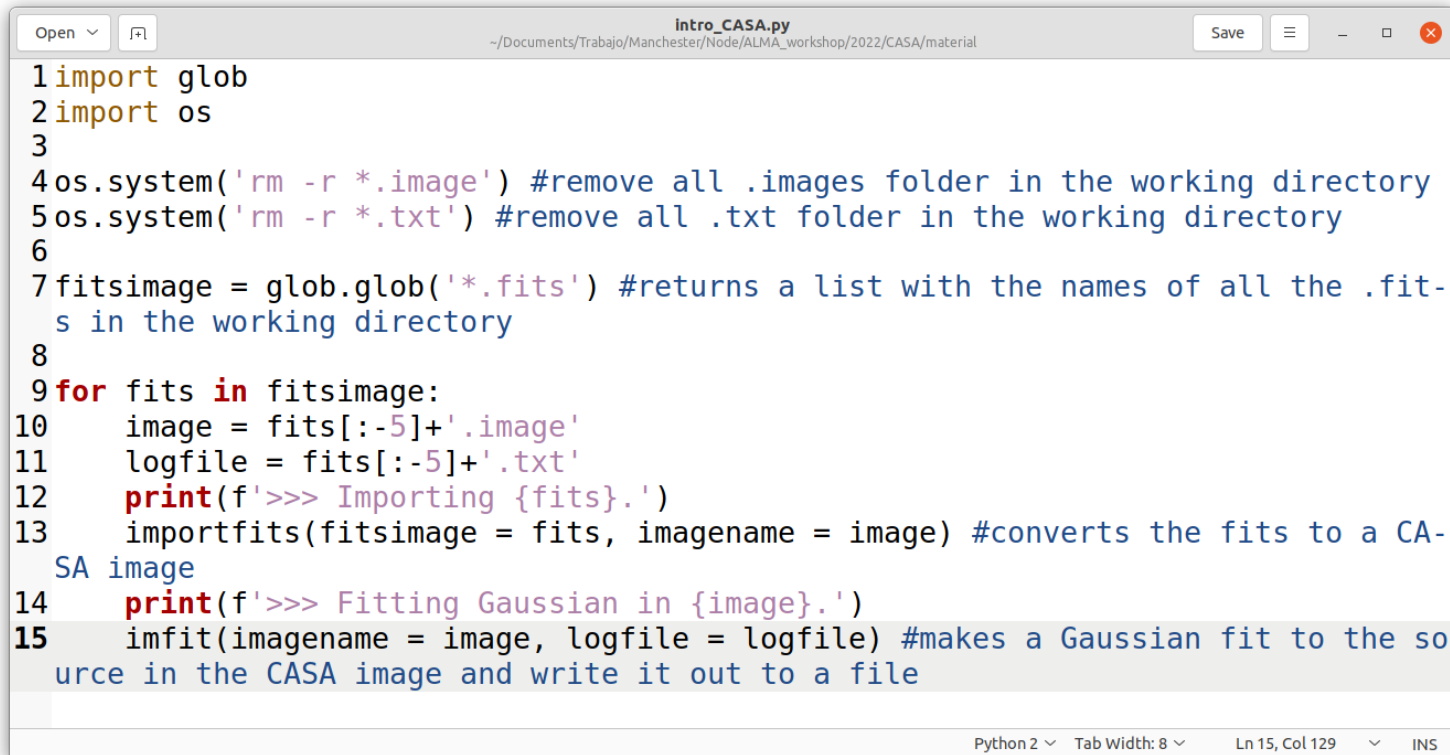
`casa -c my_script.py`

Useful when you have to process several images/sources in a similar way

* Python 3 for CASA 6.x and Python 2 for CASA 5.x and earlier

Example

- `intro_CASA.py` in the folder 'material'



```
intro_CASA.py
~/Documents/Trabajo/Manchester/Node/ALMA_workshop/2022/CASA/material

1 import glob
2 import os
3
4 os.system('rm -r *.image') #remove all .images folder in the working directory
5 os.system('rm -r *.txt') #remove all .txt folder in the working directory
6
7 fitsimage = glob.glob('*.fits') #returns a list with the names of all the .fits
  s in the working directory
8
9 for fits in fitsimage:
10     image = fits[:-5]+'.image'
11     logfile = fits[:-5]+'.txt'
12     print(f'>>> Importing {fits}.')
13     importfits(fitsimage = fits, imagename = image) #converts the fits to a CA-
  SA image
14     print(f'>>> Fitting Gaussian in {image}.')
15     imfit(imagename = image, logfile = logfile) #makes a Gaussian fit to the so-
  urce in the CASA image and write it out to a file

Python 2  Tab Width: 8  Ln 15, Col 129  INS
```

CASA Guides

Welcome to CASA Guides



CASA (Common Astronomy Software Applications) is a comprehensive software package to calibrate, image, and analyze radio astronomical data from interferometers (such as ALMA and VLA) as well as single dish telescopes. This wiki provides tutorials for reducing data in CASA.

Homepage 	Newsletter 	CASA Docs 	Download 
Helpdesk 	Subscribe 	Forum 	Tips 

CASA Tutorials



[Extracting Scripts from Tutorials](#)

Information for authors: [MediaWiki markup language](#) CASAGuides [Instructions for Authors](#)

583 articles since July 2009.

Consult the [Wiki User's Guide](#) for information on using the wiki software.