



CASA main features

- **CASA = Common Astronomy Software Applications**
- **Development started in the 90s as the next generation of AIPS**
- **Refocussed in 2003 to be *the ALMA/EVLA analysis package***
- **Has the intention to be a *general software package to reduce both interferometer and single-dish data***
- **Internally consists of two parts:**

User interface, higher-level analysis routines, viewers
= *casa non-core*



General physical and astronomical utilities, infrastructure
= *casacore*

- **Implements the “Measurement Equation” (Hamaker, Bregman & Sault 1996)**
- **Internal data format is the “Measurement Set” (Kemball & Wieringa 2000)**
- **1.5 Million lines of code (mostly C++)**
- **In public release under GNU Public License since December 2009**

CASA – development team



CASA Developers Meeting, NRAO, Socorro, May 2010

CASA – development team



Since mid 2008, two CASA developers at ESO, since Sept. 2009 three



CASA – development team

Originally only developed at NRAO (Socorro, NM), now

approx. 17 FTE developers are at work at

US (NRAO and others): 10.5

Japan (NAOJ): 3.0

Europe (ESO and others): 3.5

+ 1 CASA manager (NRAO Socorro) = Nick Elias

+ 1 Project Scientist (NRAO Socorro) = Jürgen Ott

+ a few 5% FTEs at ASTRON, ATNF, and other places

Also involved:

ALMA Computing Managers = B. Glendenning (NRAO), G. Raffi, P. Ballester (ESO)



CASA design and implementation

Overall architecture:

- 1) A data structure
- 2) A set of data import/export facilities
- 3) A set of tools for data access, display, and editing
- 4) A set of tools for science analysis
- 5) A set of high-level analysis procedures (“tasks”)
- 6) A programmable command line interface with scripting
- 7) Documentation



CASA design and implementation

Overall architecture:

1) A data structure

Tables: Images, Caltables, and the Measurement Set (MS)

2) A set of data import/export facilities

the so-called fillers: ASDM → MS, FITS → Image, UVFITS → MS, VLA → MS, etc.

3) A set of tools for data access, display, and editing

tools to load/write data into/from casacore data types,

Qt-based table browser, viewer, and (beta) x/y plotter, matplotlib-based x/y plotter

4) A set of tools for science analysis

built around the Measurement Equation (developed in 1996) = a set of C++ classes for radio astronomical calibration and imaging

5) A set of high-level analysis procedures (“tasks”)

special procedures for each required task such as CLEAN etc.

6) A programmable command line interface with scripting

Python (augmented by IPython) gives a MATLAB-like interactive language

7) Documentation

an extensive cookbook (500 pages) + documentation through help commands (help, ?, pdoc) + online help pages for users and developers



CASA design and implementation

Overall architecture:

- 1) A data structure
 - a) → *Tables: Images, Caltables, and the Measurement Set (MS)*
- 2) A set of data import/export facilities
the so-called fillers: ASDM → MS, FITS → Image, UVFITS → MS, VLA → MS, etc.
- 3) A set of tools for data access, display, and editing
tools to load/write data into/from casacore data types, Qt-based table browser, viewer, and (beta) x/y plotter, matplotlib-based x/y plotter
- 4) A set of tools for science analysis
 - b) → *built around the Measurement Equation (developed in 1996) = a set of C++ classes for radio astronomical calibration and imaging*
- 5) A set of high-level analysis procedures (“tasks”)
special procedures for each required task such as CLEAN etc.
- 6) A programmable command line interface with scripting
 - c) → *Python (augmented by IPython) gives a MATLAB-like interactive language*
- 7) Documentation
an extensive cookbook (500 pages) + documentation through help commands (help, ?, pdoc) + online help pages for users and developers



CASA design and implementation

CASA special features:

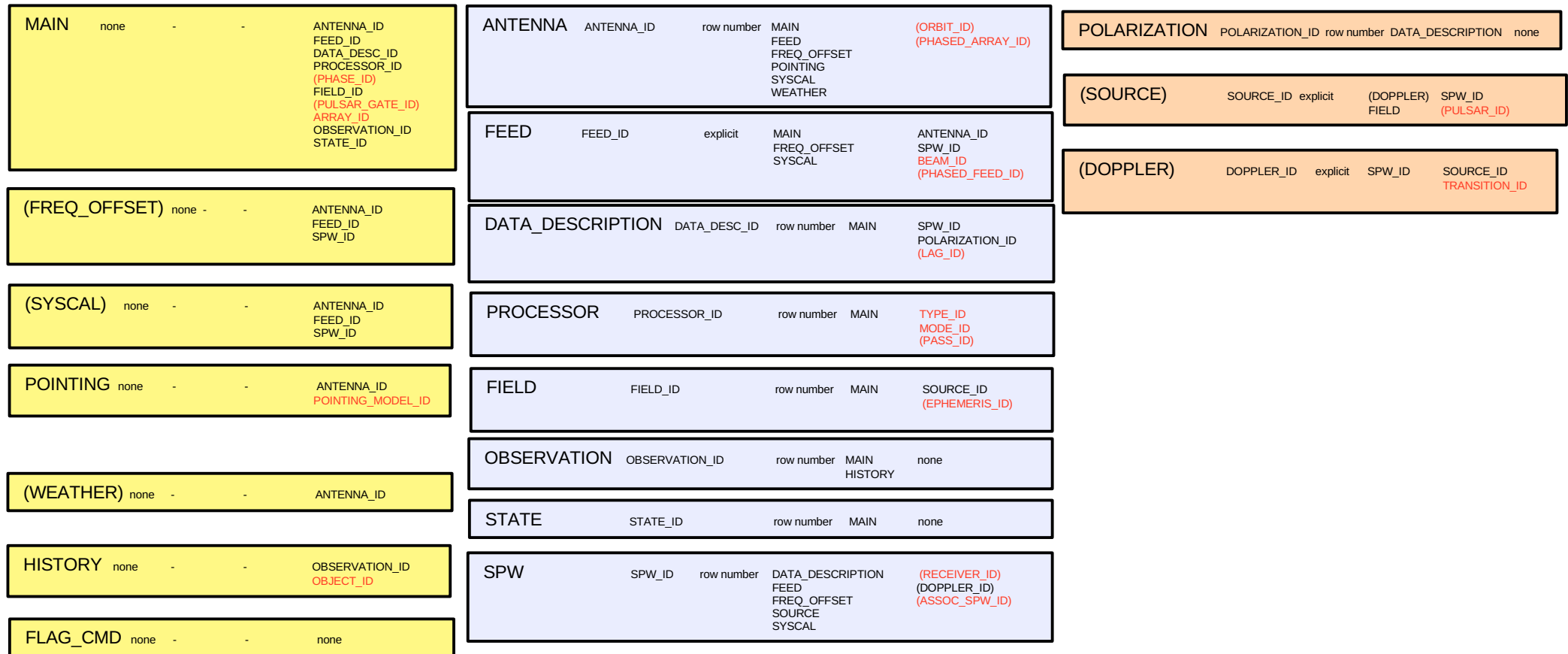
a) the *Measurement Set* (MS)

- developed by Cornwell, Kemball, & Wieringa between 1996 and 2000
- designed to store both interferometry (multi-dish) and single-dish data
- supports (in principle) any setup of radio telescopes
- supports description and processing of the data via the Measurement Equation
- fundamental storage mechanism: *CASA Tables* (inspired by *MIRIAD*)
- *MS = table for radio telescope data (visibilities) + auxiliary sub-tables*



CASA design and implementation

The Measurement Set



Legend:

[Table Name] [Key defined in this table] [key definition method] [referenced by] [referenced keys (optional)]
 reference to table outside the MS definition

Level 1: Tables not referenced by other tables

Level 2: Tables referenced by level 1

Level 3: Tables referenced by level 2



CASA design and implementation

CASA special features:

- b) the *Measurement Equation* (Hamaker, Bregman, & Sault 1996 + Sault, Hamaker, & Bregman 1996) implemented as a set of C++ classes for radio astronomical calibration and imaging

$$\vec{V}_{ij} = \vec{M}_{ij} \vec{B}_{ij} \vec{G}_{ij} \vec{D}_{ij} \int \vec{E}_{ij} \vec{P}_{ij} \vec{T}_{ij} \vec{F}_{ij} S \vec{I}_v(l, m) e^{-i2\pi(u_{ij}l + v_{ij}m)} dl dm + \vec{A}_{ij}$$

where

the vectors are: V = visibility = $f(u, v)$, I = Image to be calculated,

A = additive baseline-based error component

the matrices are: M = multiplicative, baseline-based error component

B = bandpass response

G = generalised electronic gain

D = polarisation leakage

E = antenna voltage pattern

P = parallactic angle

T = tropospheric effects

F = ionospheric Faraday rotation

S = mapping of I to the polarization basis of the observation

other variables and indices are:

l, m = image plane coordinates, i, j = telescope ID pairs = baseline, u, v = Fourier plane coordinates



CASA design and implementation

CASA special features:

- b) the *Measurement Equation* (Hamaker, Bregman & Sault 1996)
implemented as a set of C++ classes for radio astronomical calibration and imaging
(continued)

Assuming, e.g., independence of the matrices from (l,m) , the ME can be solved for individual calibration components.

$$\vec{V}_{ij}^{obs} = \vec{B}_{ij} \vec{G}_{ij} \vec{D}_{ij} \vec{P}_{ij} \vec{T}_{ij} \vec{F}_{ij} \vec{V}_{ij}^{ideal}$$

ideal visibility known from calibrator source

⇒ have set of linear equations.

The actual calculation of the component is then a χ^2 minimization.

The calibrator (cb) tool contains a set of ***solvers*** for the different calibration components.



CASA design and implementation

CASA special features:

c) A programmable command line interface with scripting

*Framework Architecture of 17 tools can be bound to any scripting language, presently selected is **Python (augmented by IPython)***

at – atmosphere library

ms – Measurement Set utilities

mp – Measurement Set Plotting, e.g. data (amp/phase) versus other quantities

cb – Calibration utilities

cp – Calibration solution plotting utilities

im – Imaging utilities

ia – Image analysis utilities

fg – flagging utilities

tb – Table utilities (selection, extraction, etc.)

me – Measures utilities

tp – table plot

vp – voltage patterns

qa – Quanta utilities

cs – Coordinate system utilities

pl – matplotlib functionality

sd - ASAP = ATNF Spectral Analysis Package (single-dish analysis imported from ATNF)

sm - simulation



CASA design and implementation

CASA special features:

c) A programmable command line interface with scripting

(continued)

Python (augmented by IPython)

Gives features such as

- tab completion
- autoparenthesis
- command line numbering
- access to OS, e.g.
 - Lines starting with '!' go to the OS.
 - `a = !ls *.py` to capture the output of `'ls *.py'`.
 - `!cmd $myvar` expands Python var `myvar` for the shell.
- history
- `execfile()`
- comfortable help



CASA design and implementation

CASA special features:

c) A programmable command line interface with scripting

(continued)

In addition to toolkit: high-level tasks for the standard user

toolkit (implemented in C++) —► tasks (implemented in Python)

e.g. the task *importfits* is based on the tool *ia* (image analysis):

```
#Python script
casalog.origin('importfits')
ia.fromfits(imagename,fitsimage,whichrep,whichhdu,zero blanks,overwrite)
ia.close()
```

CASA 3.0.1 comes with 91 implemented tasks.



CASA status

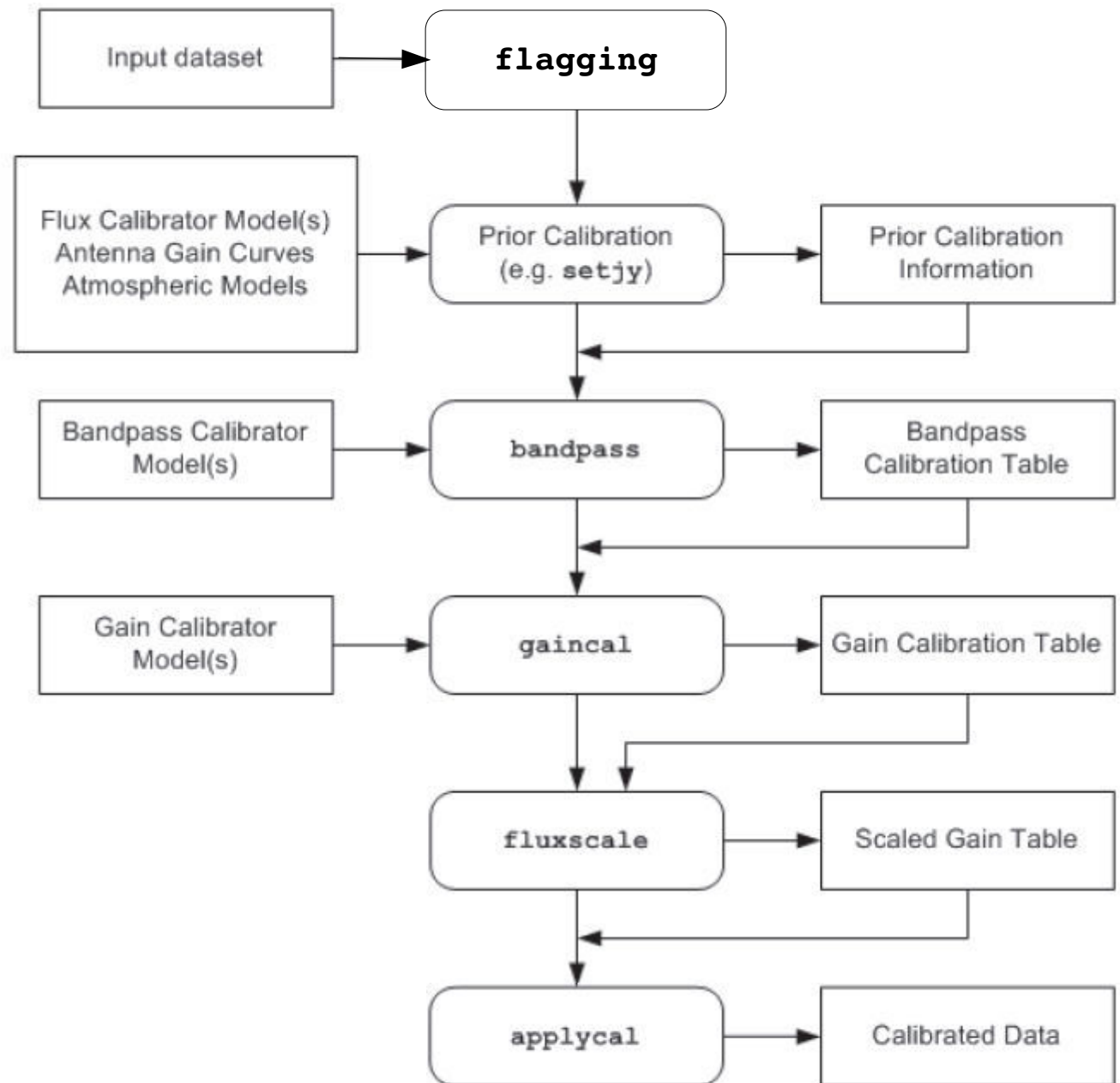
- Since Dec 2009 in public release under GPL = anybody can download,
no warranty (see <http://casa.nrao.edu>),
limited support (help desk, needs registration)
- Tutorials for the user community regularly given
- The first public release was CASA 3.0.0 (Dec 2009), release 3.0.2 published this month
- Development platforms: Linux (RHEL) + Mac OS X
- Supported platforms (binary distribution): RHEL, Fedora, openSuSE, Ubuntu, Max OS X
- Code kept in *svn* repository at NRAO, Socorro
- Presently have approx. 4300 modules, 1.5E6 lines of code, 1E6 lines of comments
- The core functionality (*casacore*, also available at <http://code.google.com/p/casacore/>)
is also used by other projects
- *Hot topics*:
 - Support for High Performance Computing and Parallelisation
 - Advanced Imaging: wide fields, continuum imaging over wide spectral ranges
 - Interoperability: using CASA for other observatories and VLBI



How does CASA look and feel?

A typical analysis session

Part 1: flagging and calibration

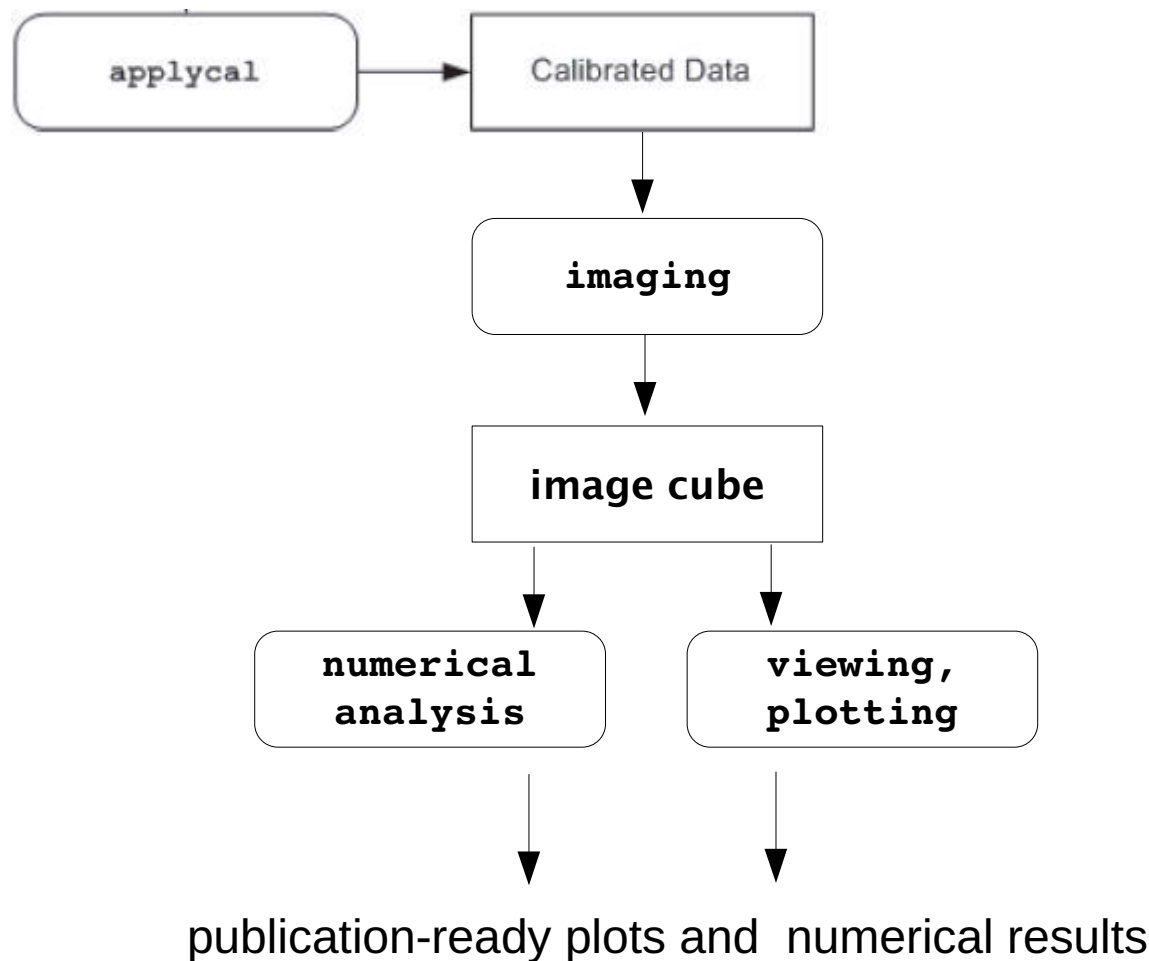




How does CASA look and feel?

A typical analysis session

Part 2: imaging and
image analysis





How does CASA look and feel?

Pictures from a typical analysis session

- 1) Startup:
open terminal and start *casapy*

Available tasks and tools are listed and the logger window is opened.

```
dpetry@M83:~/temp/casa-bologna2010
[dpetry@M83 casa-bologna2010]$ casapy
CASA Version 3.0.1 (r11099)
Compiled on: Thu 2010/04/15 04:08:39 UTC

-----
For help use the following commands:
tasklist           - Task list organized by category
taskhelp           - One line summary of available tasks
help taskname      - Full help for task
toolhelp           - One line summary of available tools
help par.parametername - Full help for parameter name
Single Dish sd* tasks are available after asap_init() is run
-----
Activating auto-logging. Current session state plus future input saved.
Filename          : ipython.log
Mode               : backup
Output logging    : False
Raw input log     : False
Timestamping      : False
State             : active

CASA <2>: 
```



The CASA user interface

The logger provides functionality for monitoring and debugging command execution.

Log Messages (M83/export/home/M83/dpetry/temp/casa-bologna2010/casapy.log)			
File Edit View			
Search Message: <input type="text"/> Filter: Time <input type="text"/>			
Time	Priority	Origin	Message
2010-04-23 12:04:03	INFO	plotms::...	##### Begin Task: plotms #####
2010-04-23 12:04:03	INFO		plotms:::casa
2010-04-23 12:04:04	INFO		plotms:::casa
2010-04-23 12:04:04	INFO	plotms::...	##### End Task: plotms #####
2010-04-23 12:04:04	INFO	plotms::...	#####
2010-04-23 12:08:11	INFO		plotxy:::casa
2010-04-23 12:08:11	INFO	plotxy::...	#####
2010-04-23 12:08:11	INFO	plotxy::...	##### Begin Task: plotxy #####
2010-04-23 12:08:11	INFO	plotxy::...	plotxy:::casa
2010-04-23 12:08:11	INFO	plotxy::t...	Switching to GUI mode. All current plots will be reset.
2010-04-23 12:08:11	INFO	plotxy::...	Adding scratch columns, if necessary.
2010-04-23 12:08:11	INFO	calibrate...	Opening MS: ah847_1-k-selected-flagged-calibd.ms for calibration.
2010-04-23 12:08:11	INFO	Calibrate...	Initializing nominal selection to the whole MS.
2010-04-23 12:08:12	INFO		Data to be selected from matches the following:
2010-04-23 12:08:12	INFO	+	Baselines: *ALL pairs of* -- VA01, VA02, VA03, VA04, VA05, VA06, VA07, VA08, VA
2010-04-23 12:08:12	INFO	+	Fields: *ALL* -- 12190+47182, 12191+48299, 1331+305
2010-04-23 12:08:12	INFO	+	Spectral Windows: *ALL* --
2010-04-23 12:08:12	INFO	+	SPW 0: *ALL Channels* -- 1 to 1 with a step of 1
2010-04-23 12:08:12	INFO	+	SPW 1: *ALL Channels* -- 1 to 1 with a step of 1
2010-04-23 12:08:12	INFO	+	Correlations:
2010-04-23 12:08:12	INFO	+	Corr. ID 0 - RR, RL, LR, LL
2010-04-23 12:08:12	INFO	+	Corr. ID 1 - *NONE*
2010-04-23 12:08:12	INFO	+	Time Range *ALL* -- 2004/5/22/01:06:05 to 2004/5/22/03:32:25
2010-04-23 12:08:12	INFO	+	Scan Numbers: *ALL* -- 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
2010-04-23 12:08:12	INFO	+	UVRanges: *ALL*
2010-04-23 12:08:12	INFO		Preparing data ...
2010-04-23 12:08:21	INFO		Now get the data
2010-04-23 12:08:23	INFO		Done Processing data ...
2010-04-23 12:08:24	INFO		Now get the data
2010-04-23 12:08:25	INFO		Done Processing data ...
2010-04-23 12:08:26	INFO		plotxy:::casa
2010-04-23 12:08:26	INFO	plotxy::...	##### End Task: plotxy #####
2010-04-23 12:08:26	INFO	plotxy::...	#####
Insert Message: <input type="text"/> <input type="button" value="+"/> <input type="button" value=""/> <input type="button" value="C"/> <input type="checkbox"/> Lock scroll			



The CASA user interface

Pictures from a typical analysis session

2) enter commands in a
MATLAB-like environment

recall previous settings

list present settings
for given task
(includes parameter
verification)

```
dpetry@pc014720:~/temp/radio-analysis/cqtau+mwc480 - Shell - Konsole
Session Edit View Bookmarks Settings Help

CASA <15>: fluxscale(vis='AT352_A071103-K', caltable='AT352_A071103-K-gain', fluxtable
='0', transfer='1')

CASA <16>: applycal(vis='AT352_A071103-K', gaintable='AT352_A071103-K-gain', field='2'

CASA <17>: tget clean
-----> tget(clean)
Restored parameters from file clean.last

CASA <18>: inp
-----> inp()

# clean :: Deconvolve an image with selected algorithm
vis                = 'AT352_A071103-K' # name of input visibility file
imagename          = 'cqtau-3-target'  # Pre-name of output images
field              = '2'               # Field Name
spw                = ''                # Spectral windows:channels: '' is all
selectdata         = False             # Other data selection parameters
mode               = 'mfs'             # Type of selection (mfs, channel, velocity,
niter              = 500               # Maximum number of iterations
gain               = 0.1               # Loop gain for cleaning
threshold          = '0.0mJy'         # Flux level to stop cleaning. Must include
psfmode            = 'clark'           # method of PSF calculation to use during min
imagermode         = ''               # Use csclean or mosaic. If '', use psfmode
multiscale         = []               # set deconvolution scales (pixels), default:
interactive        = True              # use interactive clean (with GUI viewer)
npercycle          = 100              # Number of iterations before interactive pro

mask               = []               # cleanbox(es), mask image(s), and/or region(
imsize             = [512, 512]       # x and y image size in pixels, symmetric for
cell               = ['0.03arcsec', '0.03arcsec'] # x and y cell size. default unit
phasecenter        = ''               # Image phase center: position or field index
restfreq           = ''               # rest frequency to assign to image (see help
```

The CASA user interface

Pictures from a typical analysis session

3) where needed, tools have GUIs:

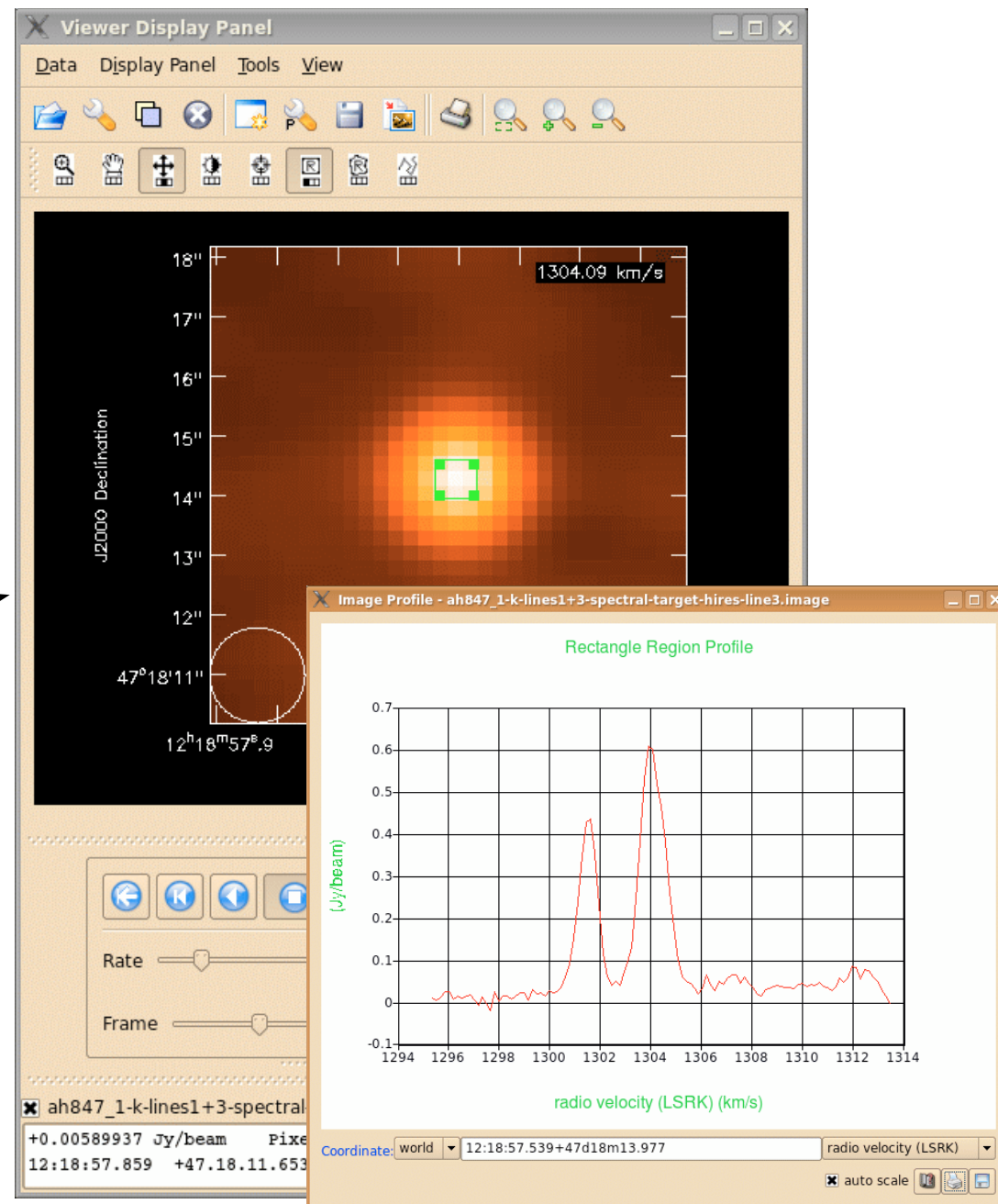
plotxy, plotcal, browsetable,
viewer, clean

(started in separate threads)

The **viewer** is a powerful multi-function tool for data selection and visualization.

Uses Qt widget set
(but 80% independent)

Rendering based on pgplot



The CASA user interface

A typical analysis session

3) where needed, tools have GUIs:

plotxy, plotcal, browsetable,
viewer, clean

(started in separate threads)

browsetable permits you to
explore any CASA table, e.g.
Measurement Sets

Also Qt-based.

Table Browser

FileEditViewToolsExportHelp

</

The CASA user interface

A typical analysis session

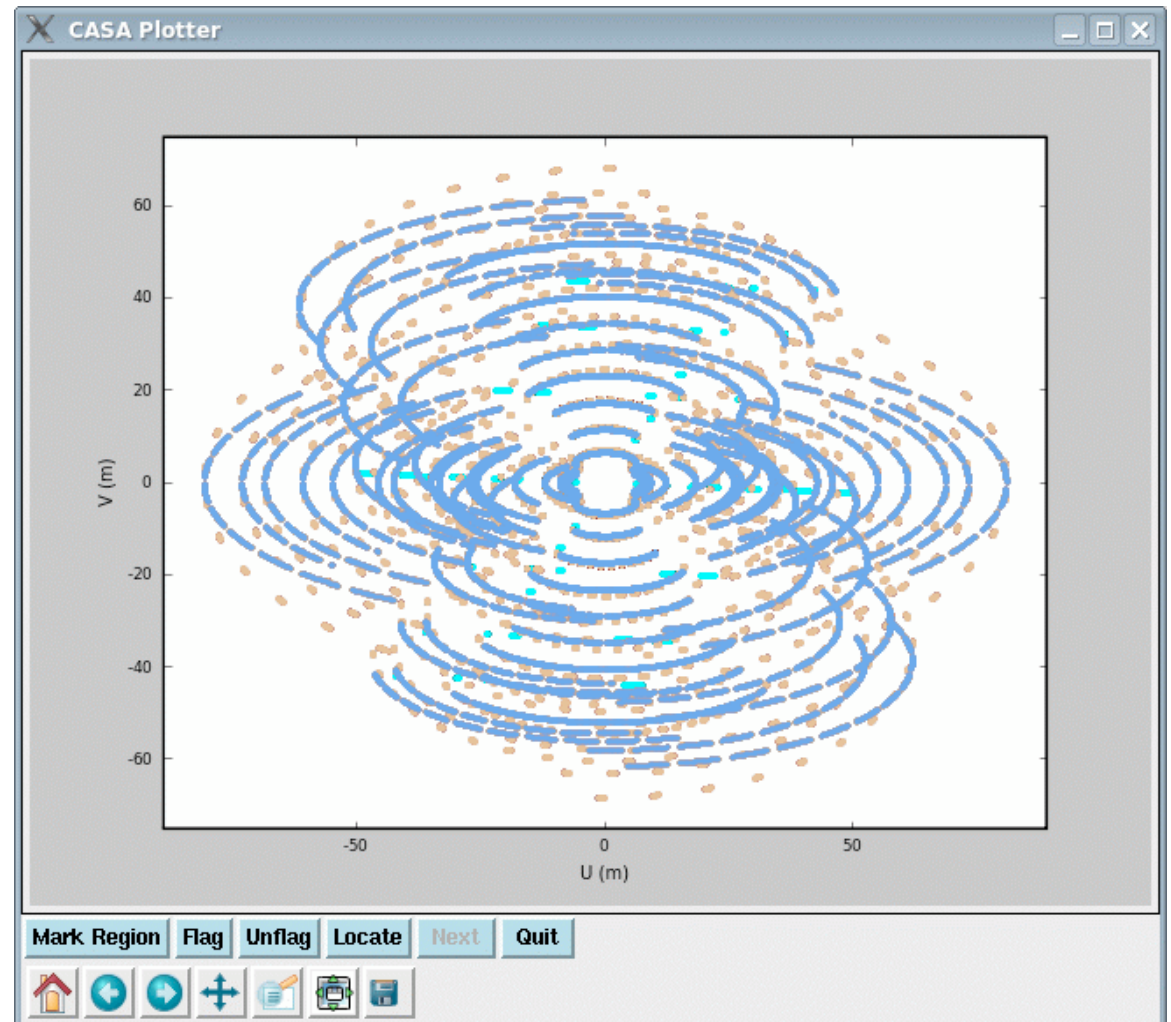
3) where needed, tools have GUIs:

plotxy, plotcal, browsetable,
viewer, clean

(started in separate threads)

plotxy is a specialized tool
for diagnostic plots and
data selection

To be phased out.



The CASA user interface

A typical analysis session

3) where needed, tools have GUIs:

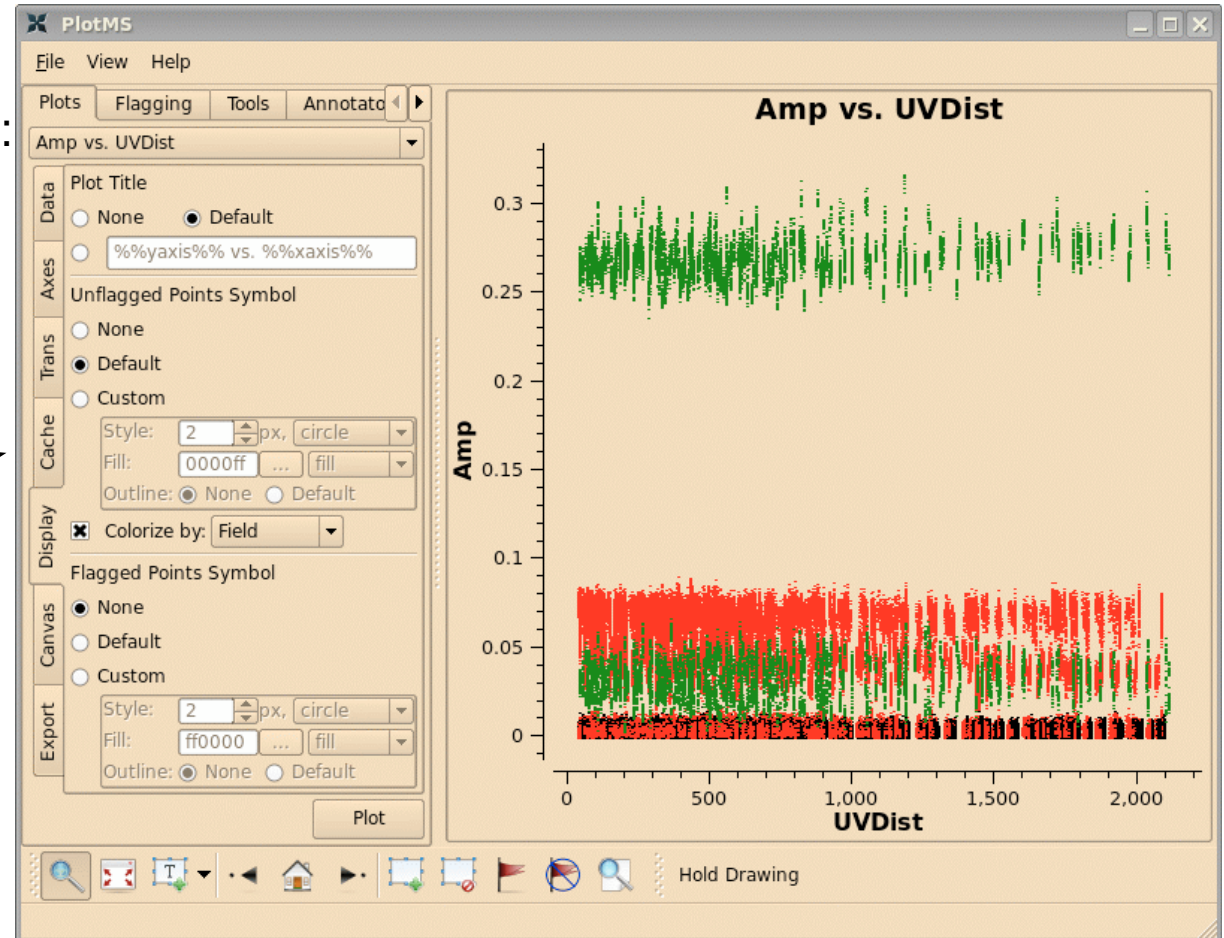
plotxy, plotcal, browsetable,
viewer, clean

(started in separate threads)

plotms is going to replace
plotxy. Release 3.0.2
contains beta version.

plotms is Qt-based and much
faster than plotxy.

Uses generic plotting class
which in turn uses **Qwt**.





Summary

- The standard science data analysis package for ALMA and EVLA is **CASA**
- Data from other observatories can also be processed, e.g. VLA, BIMA, ATCA, ...
- CASA derives from AIPS++ (partially survives in **casacore**)
- approx. 20 people are working on CASA
in North America, Europe, and Japan
- CASA is a **toolbox** with
 - MATLAB-like user interface
 - GUI tools for data selection, browsing, and image processing
- the heart of the science analysis code is the **Measurement Equation**
- the internal data format are **CASA Tables**
- the **Measurement Set** is the CASA data format for visibility data
(it is technically a Table with several well-defined sub-tables)
- CASA is publicly available under GPL for **Linux and Mac OS X**

- The first public release of CASA (version 3.0.0) became available in December 2009
- The latest release is version 3.0.2



Additional remarks

Dirk Petry (ESO), June 2010



CASA Measurement Sets, ASDMs, and uvfits

- Internal CASA visibility data format is the **Measurement Set (MS)**
- Presently supported input formats:
 - ALMA: **ALMA Science Data Model (ASDM)**
 - EVLA: Science Data Model (SDM, same as the ASDM)
 - VLA: **VLA archive format**
 - FITS IDI: **planned for later this year**
 - and the transport format **uvfits**



CASA Measurement Sets, ASDMs, and uvfits

The **MS**

- relational database system with fixed structure made from *CASA Tables*
- consists of a main *table* with 15 required *sub-tables* + several optional ones
- uses OS directory structure (need to copy with `cp -R`, remove with `rm -r`)
- visibilities stored in the MAIN table
- no compression
- manipulate an MS with the *ms* and the *tb* tool or with *browsetable()*
- during processing, CASA may add “scratch columns” to the MS main table



CASA Measurement Sets, ASDMs, and uvfits

The **ASDM**

- relational database system with fixed structure
- consists of set of up to 56 tables (also observatory setup information!)
- uses OS directory structure (need to copy with `cp -R`, remove with `rm -r`)
- visibilities stored in the MAIN table
- no compression
- on disk, table descriptions in XML files, table data in binary MIME format files
- import into CASA using the task *importasdm* (for v1) or *importoldasdm* (for v0.9)
- in release 3.0.2 there is a first version of *exportasdm* (MS to ASDM)



CASA - further information

Further information on CASA

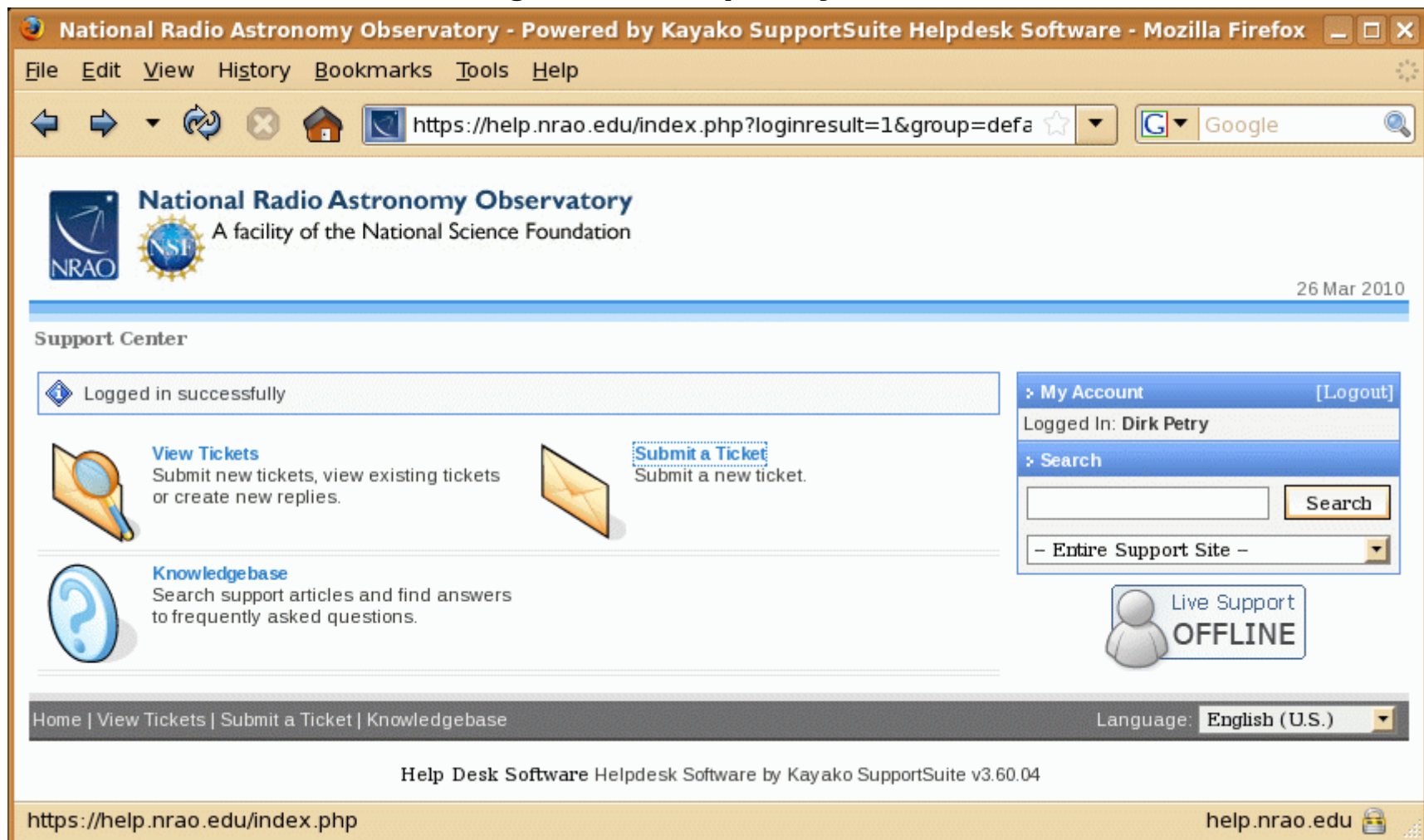
- CASA home page <http://casa.nrao.edu>
- CASA release notes http://casa.nrao.edu/release_notes.shtml
- CASA cookbook http://casa.nrao.edu/Doc/Cookbook/casa_cookbook.pdf
- CASA guides <http://casaguides.nrao.edu>

In case of problems ...

What to do if you encounter a problem with CASA:

If the cookbook and the release notes don't help, go to <http://help.nrao.edu/>

Don't have an account? Register at <http://my.nrao.edu>

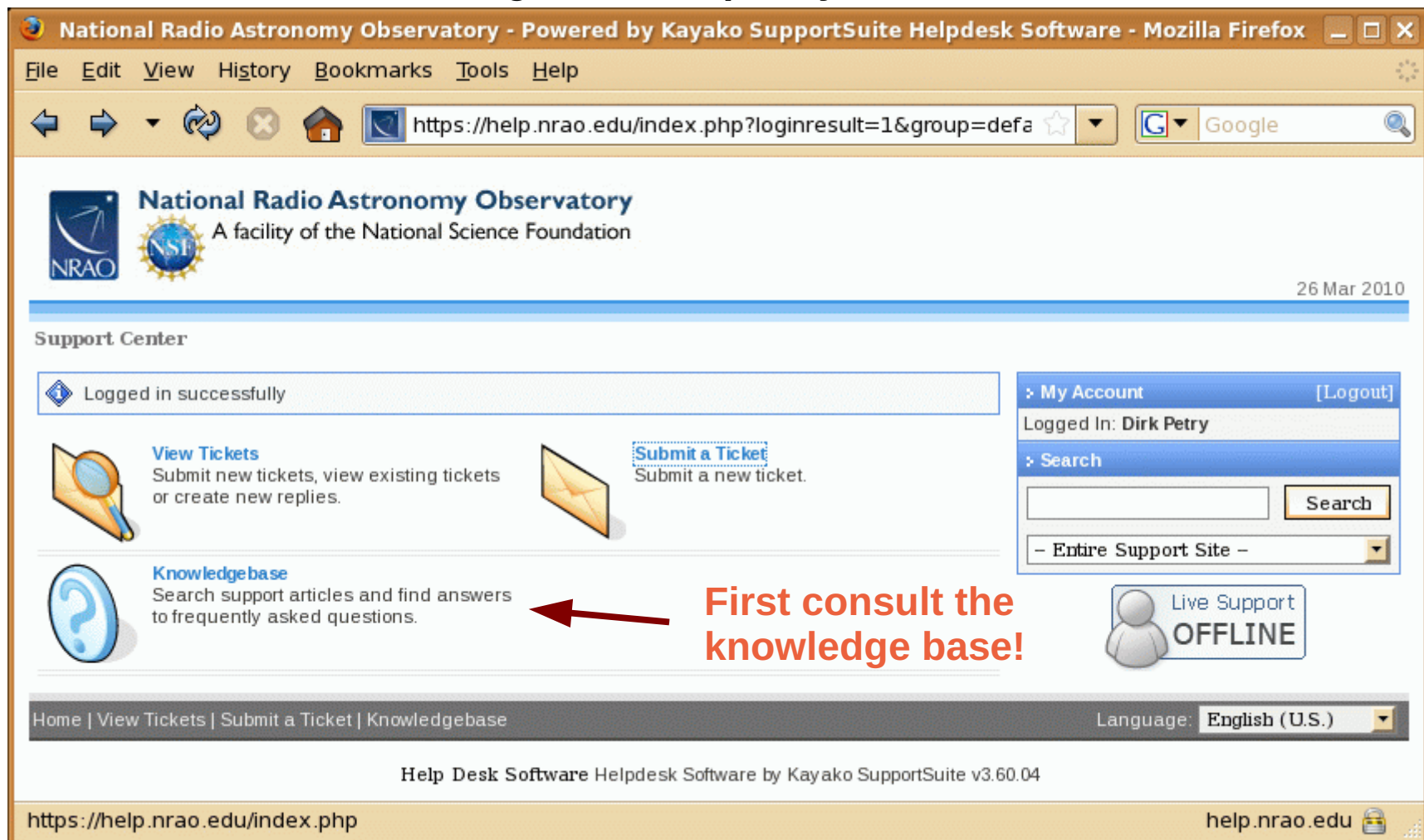


In case of problems ...

What to do if you encounter a problem with CASA:

If the cookbook and the release notes don't help, go to <http://help.nrao.edu/>

Don't have an account? Register at <http://my.nrao.edu>





In case of problems ...

**What to do if you encounter a problem with CASA
and you can't find the solution in the documentation or the knowledge base:**

A) You think you might have found a bug in CASA

- Try to reproduce your problem, ideally by writing a Python **script** which will demonstrate the problem.
- Put your **test data (if needed) on some web or ftp server** where it can remain for at least several months.
- File a helpdesk ticket including the script, a short description of the problem, and the URL of the data.
- Need to mention **CASA version** and your **operating system (32 bit or 64 bit?)**

B) You don't know how to perform a certain analysis task in CASA

- If you can't make progress, then, as in (A) try to prepare a **script** for your analysis up to the point where you don't know how to go further.
- File a helpdesk ticket including the script and a description of what you would like to achieve.

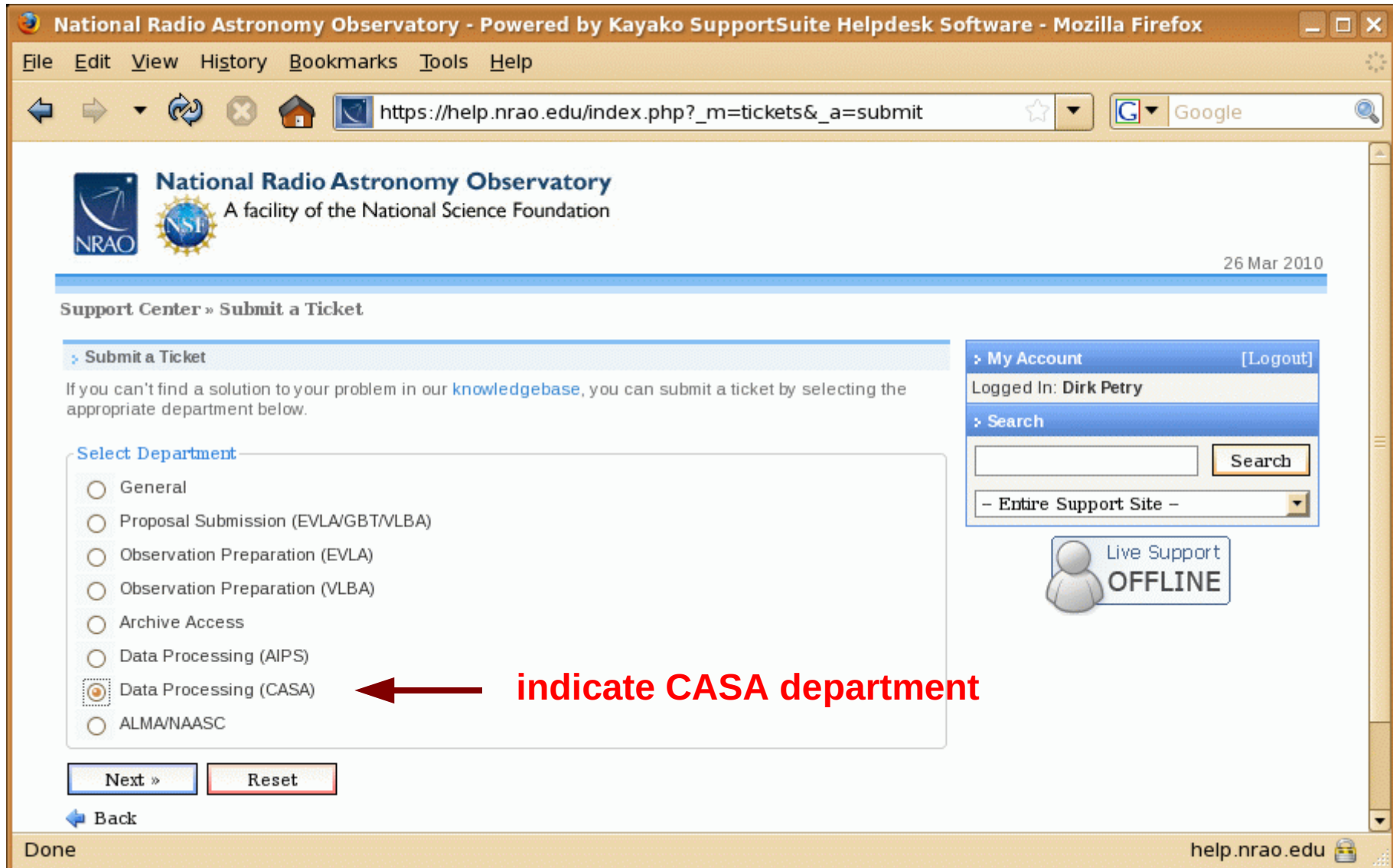


In case of problems ...

How to file a helpdesk ticket at help.nrao.edu:

In case of problems ...

How to file a helpdesk ticket at help.nrao.edu:



National Radio Astronomy Observatory - Powered by Kayako SupportSuite Helpdesk Software - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://help.nrao.edu/index.php?_m=tickets&_a=submit

NRAO National Radio Astronomy Observatory
A facility of the National Science Foundation

26 Mar 2010

Support Center » Submit a Ticket

Submit a Ticket

If you can't find a solution to your problem in our [knowledgebase](#), you can submit a ticket by selecting the appropriate department below.

Select Department

- ☐ General
- ☐ Proposal Submission (EVLA/GBT/VLBA)
- ☐ Observation Preparation (EVLA)
- ☐ Observation Preparation (VLBA)
- ☐ Archive Access
- ☐ Data Processing (AIPS)
- ☒ Data Processing (CASA) ← indicate CASA department
- ☐ ALMA/NAASC

Next » Reset

Back

My Account [Logout]
Logged In: Dirk Petry

Search

Search

- Entire Support Site -

Live Support
OFFLINE

Done help.nrao.edu



In case of problems

How to file a helpdesk ticket:

Where does **your data** come from?
(identify necessary expertise)

Where do **you** come from?
(who is responsible?)

What **OS and CASA version** are
you using?
(for reproducing your problem)

Give at least a **description** of what
you are trying to do and the **URL of
your test data** if needed to reproduce
your problem.
Also **quote error messages**.

Upload a Python **script** which
demonstrates your problem

National Radio Astronomy Observatory - Powered by Kayako SupportSuite H

File Edit View History Bookmarks Tools Help

Priority: Default

CASA

Data Source: *
The telescope that produced the data you are analyzing

☐ EVLA
☐ ALMA
☒ Other (give telescope in details)

Region: *
☐ North America
☒ Europe
☐ East Asia
☐ Other

Operating System: *
The operating system on which you are running CASA

☐ RedHat 4-32bit
☐ RedHat 5-32bit
☐ RedHat 5-64bit
☐ Other Linux 32-bit (give type in details)
☐ Other Linux 64-bit (give type in details)
☒ Mac Intel 10.5
☐ Mac Intel 10.6

Version: *
CASA version you are using (e.g., 3.0.0; can be obtained from first line of logger messages after startup)

3.0.0

Message Details

Subject: * possible bug in importvla

I am trying to analyse VLA data.
A sample dataset can be found at <http://myinstitute.org/~myself/mydata.tgz>
With the attached script I get the error message

SEVERE: error in importvla - cannot do this and that

Knowledgebase suggestions

No relevant knowledgebase articles found.

Upload File(s)

Browse...

Done help.nrao.edu

In case of problems ...

National Radio Astronomy Observatory - Powered by Kayako SupportSuite Helpdesk Software - Mozilla Firefox

File Edit View History Bookmarks Tools Help

https://help.nrao.edu/index.php

National Radio Astronomy Observatory
A facility of the National Science Foundation

26 Mar 2010

Support Center » Submit a Ticket » Data Processing (CASA)

Submit a Ticket

Your ticket has been submitted to our department successfully. One of our support agents will get back to you with more information shortly.

Ticket Information

Ticket ID:	CBM-918027
Department:	Data Processing (CASA)
Full Name:	Dirk Petry
E-mail:	dpetry@eso.org
Priority:	Default

CASA

Data Source: The telescope that produced the data you are analyzing	ALMA
Region:	Europe
Operating System: The operating system on which you are running CASA	RedHat 5-32bit
Version: CASA version you are using (e.g., 3.0.0; can be obtained from first line of logger messages after startup)	3.0.0

My Account [Logout]
Logged In: Dirk Petry

Search

Search

- Entire Support Site -

Live Support
OFFLINE

your ticket ID

(confirmation email with ID in subject will arrive from do-not-reply@nrao.edu)

Done help.nrao.edu