Geneva Astronomical Data Centre

ASTRO_IMAGE Users Manual

Reference	:	astro_image
Issue	:	1.1
Date	:	14 March 08

Geneva Astronomical Data Centre
Chemin d'Écogia 16
CH–1290 Versoix
Switzerland

http://isdc.unige.ch/index.cgi?Soft+astroroot

Authors and Approvals

AstroROOT	**astro_image Users Manual**	
14 March 08	1.1	astro_image 1.1

Prepared by : R. Rohlfs

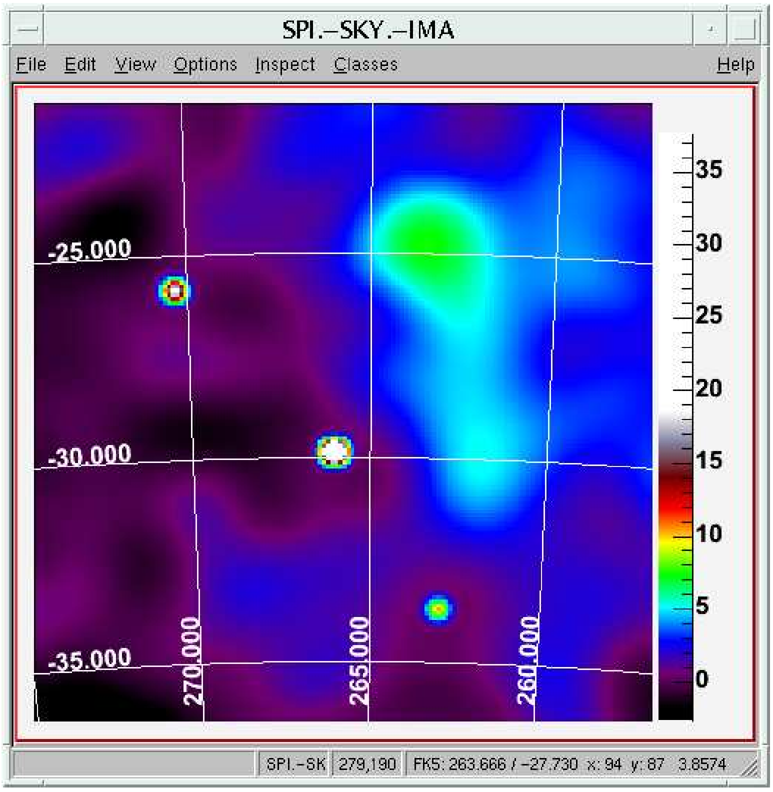
Document Status Sheet

AstroROOT	astro_image Users Manual	
27 APR 04 14 MAR 08	1.0 1.1	astro_image Users Manual released use ISDC_ENV instead of WORK_ENV
14 MAR 2008	Printed	

Contents

1 Introduction

The astro_image component is part of the AstroROOT - package. It is a C++ class to display astronomical images. As other display tools of ROOT the astro_image display uses the TPad as background to display the image. astro_image is derived from the standard ROOT - class TASImage and uses most of the freatures of this ROOT class. These are specially the drawing function the zooming capability and the palette editor. The new functions of astro_image compared to TASImage are reading FITS files, drawing astronomical coordinate systems and the handling of regions including reading and writing region files.



2 How to Build

To build this astro_image class the full AstroROOT package has to be downloaded and build. The tf_container user manual describes in detail the procedure to install the AstroRoot package. After a succesfull installation the AstroImage class will be part of the libastr.so library and the astro_image program will be installed in the ${ISDC_ENV}/bin - directory.

3 astro_image - constructor

The AstroImage class has three constructors to display an astronomical image:

```
AstroImage(const char * filename, const char * name, UInt_t cycle);
```

Opens and displays and image. It can read FITS files (*.fits), ROOT files (*.root) and Asro - files (*.asro). See the tf_container user manual for more information about these file formats.

name is the extension name of the image in the file. This parameter is optional (set to NULL or "") for FITS files and mandatory for ROOT and ASRO files. If it is defined for FITS files if must match the extension name of the specified extension or the parameter cycle must be set to 0.
The parameter cycle is the extension number for FITS Files. Note: the primary header has cycle number 1 and the first extension is cycle number 2! This is the cycle number for a given name for ASRO and ROOT files. Note in these two file formats each extension name has its own counter for cycle numbers. There can be easily several images with cycle number 1 in the same ROOT (ASRO) file if they have different names.

```
AstroImage(TFBaseImage * image);
```

displays an already opened image, opened by using the tf_container classes. The parameter image can be any derived class of TFBaseImage and therefore it can display images of every data type from short integer to double.

```
AstroImage(fitsfile * fptr);
```

displays an already open image, read with cfitsio functions.

4 Draw an image

To draw an image with this image class the member function Draw() has to be called, like for all other drawing classes in ROOT. For example:

```
AstroImage * img = new AstroImage(fileName, imageName, extension);
img->Draw();
```

To display the astronomical coordinates, the pixel coordinates and the pixel value of the pixel the mouse is pointing to in the status bar call either

```
win->ToggleEventStatus();
```

of the TCanvas (win) the image is display or select the Menu Options->Event Status of the TCanvas - window.

5 Image - Context Menu

The context menu of the astro_image class provides several features, like zooming, coordinate system, region and region files, style of the coordinate system and coordinate labels. All are described in more details hereafter.

5.1 Coordinate System

The astro_image class uses the wcs - library to convert the coordinates into different systems and to calculate the actual position of the coordinate lines on the image. To change the drawn coordinate system the menu item **CoordinateSystem** should be used. Following coordinate systems are supported: ICRS, FK4, FK5, PHYSICAL, B1950, J2000, GALACTIC, ECLIPTIC and LINEAR. The function CoordinateSystem() checks only the first characters of the input string until it is

unequivocal and it is not case sensitive. For example following inputs would be accepted: "i", "fk4", "FK5", "p", "b", "j", "G", "E" and "l".

The labels can be expressed in degree and fraction of degree or in degree (hour), minutes and seconds. To toggle between these modes the menu item **SetDeg** must be selected.

The astro_image class tries to find a good number of coordinate lines and a good distance between two lines, respectively. But the user can increase and decrease the number of coordinate lines if the default value does not satisfy: Use the menu items **IncFirstAxis, DecFirsAsis, IncSecondAxis** and **DecSecondAxis**. The menu item **ReseAxis** resets the number of axis to its default value.

With the menu items **SwapFirstAxis** and **SwapSecondAxis** the labels of a coordinate line can be drawn at the other end of the line.

5.2 Zoom and Unzoom

An image can be zoomed either by the **Zoom** - item of the context menu or by defining an rectangle area with the mouse, while the left mouse button is pressed. The Zoom() - function has four parameters, defining the offset of the zoomed area from the lower left corner and the size of the zoomed area. Note: The values are always relative to the original image and in pixel coordinates, even if the image is already zoomed.
To unzoom an image the **UnZoom** - item can be used.

5.3 Palette Editor

The menu item **StartPaletteEditor** opens a GUI to edit the palette of the image. A histogram of the whole image is displayed to be able to select the intensity range the palette should be applied to. Once a palette is defined it can be saved either in a ROOT file or in an ASCII file.
Note: The palette editor is not part of the astro_image class but is part of the standard ROOT distribution.

5.4 Style of the Coordinates and Labels

The color, the text size, the thickness of the coordinate lines and the style of the coordinate lines can be modified by the standard ROOT GUIs, which pop-up with the context menus **SetLineAttributes** and **SetTextAttributes**. It is even possible to move the labels of the coordinates outside of the image by selecting option 31 of the text attribute GUI. The default position is option 11.
To display the image without coordinates the toggle menu item **SetCoordinates** must be selected.

5.5 Regions

Regions can be defined interactiverly or can be read from a region file. (See A Region File Format). The style, the size and the position of a regions can be modified with the mouse. But not all attributes are saved in the region file with the **WriteRegion** munu item.

A new text region can be created with the **AddText** menu item. Actually a TLatex is created and Latex syntax can be used.

Other regions can be created with the **AddRegion** item. These are:

- circle
- ellipse
- line
- point
  it can be
  - a cross
  - a x
  - a box
  - a diamond
  - a circle

  To change from the default circle to an other point type the **SetMarkerAttributes** menu item of the point - region has to be selected.

The text of a region is always displayd above the region with the same color as the region. This propery cannot be changed.

5.6 Title

A title can be defined with he menu item **SetTitle**. It will be displayed in a TPaveText on top of the image.

6 astro_image - Program

The astro_image program is a wrapper for the AstroImage class to display one image. The executable is like a FTOOL with a parameter file. To run the programs the environment variable PFILES must be set to the directory of the parameter files or the parameter files *.par must be in the local directory.
Noty: during installation the parameter files are installed in the directory ${ISDC_ENV}/pfiles. Therefore the easiest is to define

```
setenv PFILES ${ISDC_ENV]/pfiles
```

More information about paraemter files can be found in the PIL user manual ${ISDC_ENV}/help/pil.ps

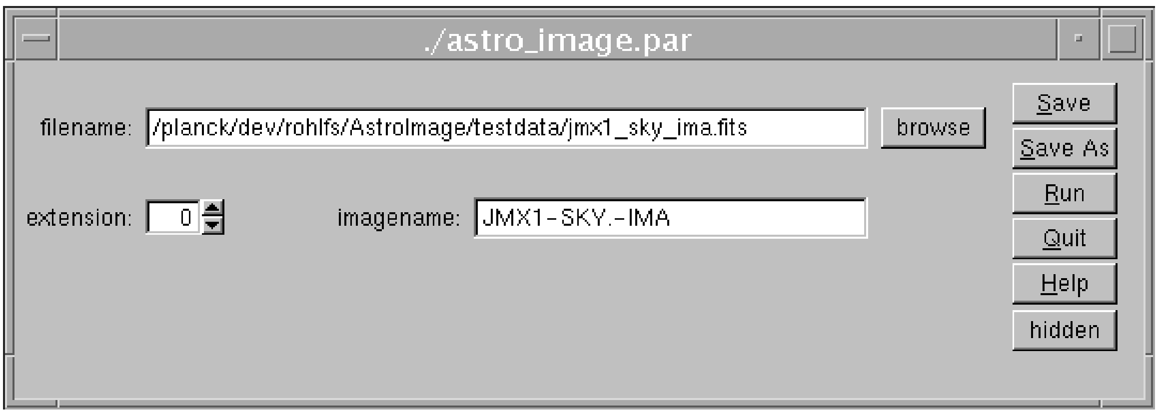
The GUI to edit the program parameters is launched if the program is called with the flag –g (minus minus g) or –gui (minus minus gui). Without this flag the program asks for the parameters. More information about the program parameters are in a help text of the program (astro_image.txt). This help text can be displayed with the Help button of the GUI.

The program has three parameters:

- filename
  it can be either a FITS file with a FITS image, a ROOT file with a TFImage class or an ASRO file with a TFImage class.
- extension
  defines the extension in FITS files or the circle number in ROOT and ASRO files. The primary array in a FITS file has extension number 1 and the first extension is number 2. In ROOT and in ASRO files each image name has its own cycle counter. Therefore several images can have the same cycle number in a ROOT or ASRO file if they have different names. If it is set to 0 the first image in the file with the defined image name is displayed.
- imagename
  is the name of the image or the extension name in FITS files. It can be an empty string for FITS files, but must be defined for ROOT and ASRO files.

To pop-up a GUI to enter the program parameter the program has to be started with the option –g or –gui:

```
astro_image --g
```



A Region File Format

The file format of the region files follow the region files of the DS9 program:

http://hea-www.harvard.edu/RD/ds9/ref/region.html

but astro_image do not support all regions and all properites of the DS9 program

Following regions are supported by AstroImage:

- circle
- ellipse
- line
- text
- circle point
- box point
- diamond point
- cross point
- x point

Following properties are supported by AstroImage:

- text
- color
- textangle
- line
- point