

IDP3

Introduction

Image Display Paradigm #3 (IDP3) is an interactive program for image visualization, manipulation, and analysis. It is written in the Interactive Data Language (IDL) developed by Research Systems, Inc. The program was initially developed in 1997 for the Near Infrared Camera and Multi-Object Spectrometer Instrument Definition Team (NICMOS IDT) to support image analysis and point spread function subtraction for high precision photometry and astrometry. Over the years the program has evolved into an extensive collection of image analysis tools capable of supporting many types of 2-D astronomical imagery.

Some of the image sources for which IDP3 has supported image analysis include:

- *Hubble Space Telescope*: NICMOS, STIS, WFPC2, ACS
- *Spitzer Space Telescope*: IRAC, IRS, MIPS
- Keck II Adaptive Optics
- Palomar/PALAO
- Canada France Hawaii Telescope
- Monolithic Mirror Telescope: Aries
- Very Large Telescope: UVES
- Solar and Heliospheric Observatory: C2 Coronagraph
- *Transition Region and Coronal Explorer Spacecraft*

The program is inherently interactive and has no “Command Line” or “Batch Mode”. All functions are selected from menus or buttons in the various widgets. Many control parameters for individual functions are set in a preferences text file [IDP3_Preferences]. This allows the user to define certain control parameters for every IDP3 session. Some of these control parameters may be reset dynamically during an IDP3 session in the Edit Preferences Widget.

Primary Features of IDP3

- a facility to work with a single image or collection of images simultaneously. Images or lists of images in **FITS**, **HDF**, **PICT**, and **TIFF** format may be loaded into IDP3. These images may be turned **ON** or **OFF** for inclusion or not in the current display {Show Images Widget}. Masks for the individual images in **FITS**, **HDF**, **PICT**, and **TIFF** formats may also be loaded, edited, and turned **ON** or **OFF** [Image Mask Builder Widget].
- Data may be aligned [translation and rotation] by visual or statistical [world coordinates or centroid solutions] criteria.
- Image combination, convolution, blinking, mask definition, and source catalog overly are available from the main display window.
- Supported arithmetic operations on images include: addition, subtraction, multiplication, division, inversion (1/image), averaging, minimization, and absolute value [Show Images Widget].
- Images may be translated, scaled, rotated, biased, resampled, padded, clipped, and rectified [Adjust Position Widget].
- Image display is controlled by scaling (plotmin & plotmax), form (linear, log, or square root), and color table selection [Adjust Display Widget].
- The Region of Interest (ROI) provides a tool for intensive examination of sub-regions of the main display. Cross sections, radial profiles, noise profiles, histograms, contour maps, image statistics, spreadsheets, aperture photometry, gaussian profile fitting, photocentric moment solutions, and surface plots may be performed from the ROI.
- Individual images may have associated masks as well as the ROI may have its own composite mask. Pixels of a specific value may be masked on-the-fly. Not a Number (NaN) values are automatically masked when data are loaded.
- Data may be resampled by bi-linear, apodized bi-cubic, or spline bi-cubic interpolation or pixel replication with the option to conserve flux. Sub-pixel image translation is done by bi-linear or apodized bi-cubic interpolation.
- All modifications to the individual images that comprise a saved output image are documented in its header.

- FITS (primary header and data unit, PHDU, multi-extension image, HDF, TIFF (2-D and 3-D), and PICT images are supported on input. FITS (PHDU and single image extension) are supported on output.
- The state of all loaded images may be saved to a parameter file which may be restored in IDP3 in a later session.
- Pixels in individual images may be edited or repaired [Image Edit and Bad Pixel Repair Widgets].

The Paradigm

The main window of IDP3 may be viewed as a palette or work area for the display and manipulation of one or more images. As images are loaded into IDP3 they are combined with the palette. Each image has its own control panel [Adjust Position Widget] for manipulation of the image within the work area by translating, scaling rotating, resampling, etc the image. A second panel [Show Images Widget] controls which images are combined in the work area and by what operation: addition, subtraction, multiplication, etc. A sub-region of the work area may be extracted for a more detailed view where functions such as radial profiles, cross sections, aperture photometry, etc. may be performed.

The methodology of IDP3 is to store each image as loaded into the primary data structure, info. As the user modifies parameters for the work area or the individual images the work area is rebuilt using the original data with the current list of requested parameters and operations. The advantage to this method is that no operations must be undone to the images. The resultant work area is saved permanently to disk only when the user specifically saves it [Main Window File Menu]. The penalty of this methodology is the time required to reprocess all of the ON images to dynamically build the current display. This has not posed a serious problem to date.

Requirements

The user must have IDL Version 5.6 or higher to run IDP3. To run IDP3 using the IDL Virtual Machine (no IDL software license required) the user must have IDL Version 6.0 or higher. IDL may be acquired from Research Systems, Inc. (<http://www.rsinc.com>). IDP3 makes use of routines in the IDL Astronomy User's Library available at no charge from the Goddard Space Flight Center at <http://idlastro.gsfc.nasa.gov>. Also, more general IDL utilities

by the NICMOS programming staff may be found in the `ua_lib` directory of the NICMOS software distribution. These libraries are retrieved separately from IDP3. The IDL utilities developed by David Fanning, Craig Markwardt, Don Lindler, and members of the JHU Applied Physics Laboratory Staff that are used by IDP3 are included in the IDP3 distribution.

Platform Support

IDP3 runs on multiple platforms with varying display resolutions and various CPU configurations. The most common platforms are: Sun Solaris, Linux, Apple Mac OSX, and MS Windows.

Setup for IDP3

Make sure that the `IDL_PATH` is defined to include all RSI IDL procedures, the `astron` library from GSFC, the `ua_lib` procedures, and the procedures of IDP3. To insure that IDL has access to all of the needed procedures the user start IDL and do the following:

```
IDL> @doidp3
```

```
IDL> resolve_all
```

If IDL cannot find any of the procedures needed for IDP3 the names will be reported on the display. If there are no missing routines the user is ready to type IDP3 to view the Main Display Widget.

Using the IDL Virtual Machine

If the user wishes to use IDP3 without a software license for IDL, he must acquire IDL (Version 6.1 or greater) from the RSI web site. IDL can be operated in demo mode which does not allow any form of output and only runs for 7 minutes. When totally resolved save sets of major procedures (as IDP3) have been built the user may run these save sets with the following command:

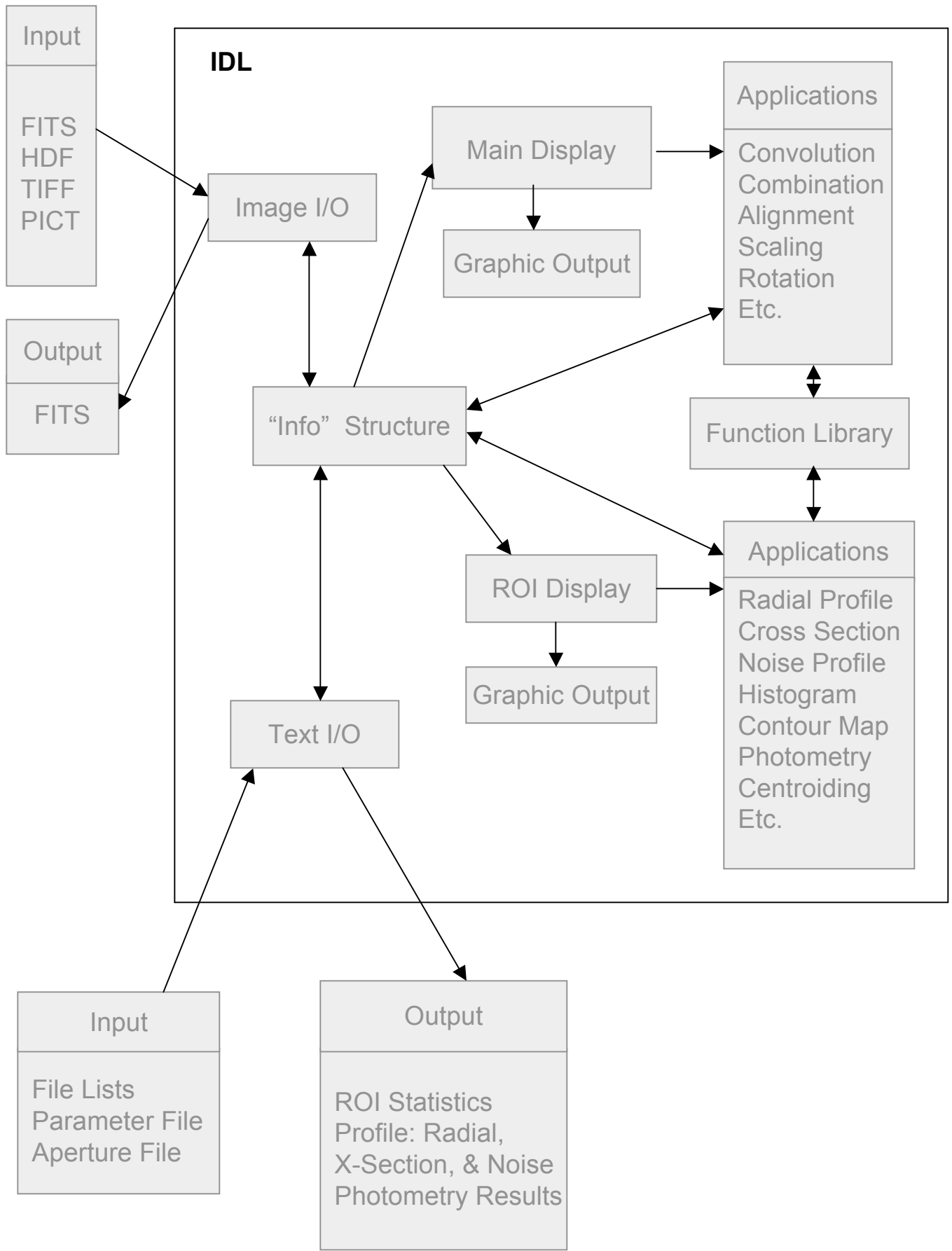
```
idl -vm = idp3.sav
```

There will be no command line interface to IDL and all output text is stored in an internal IDP3 array that may be viewed with the `Create Text Window` function in the File Menu of the Main Display.

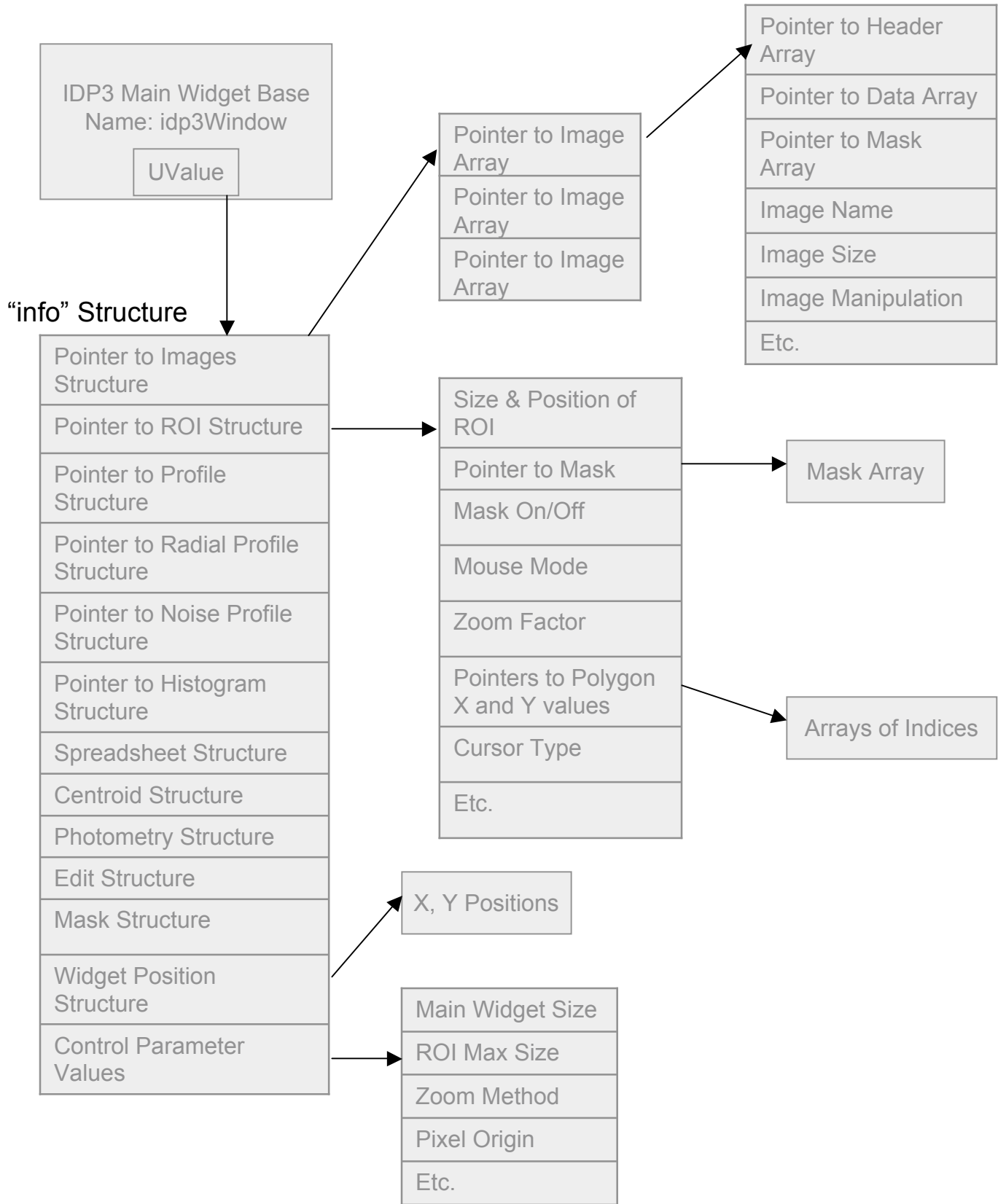
Illustrative Diagrams

The following 2 pages contain illustrations of IDP3. The first, the IDP3 Block Diagram, shows the program flow in IDP3. The second, Partial IDP3 INFO Structure, illustrates an abridged layout of IDP3's main information structure, info.

IDP3 Block Diagram



Partial IDP3 INFO Structure



Getting Started with the Preferences File

Many parameters for IDP3 that control how data are processed are set in the file, IDP3_Preferences. Most of the values can be modified on-the-fly with the Edit Preferences Widget while others may be changed in the widgets in which they are used. In the case of specifying a YES or NO value for a parameter, 0=NO and 1 = YES. Parameters that may be changed on-the-fly in the Edit Preferences Widget are noted to be changeable on-the-fly. A description of the parameters in the preferences file and their meanings follow below:

RESAMPLING: Data may be resampled in both the Main and ROI displays with a separate parameter for defining the method for each. IDP3 supports four methods for resampling the data which are changeable on-the-fly::

| | |
|---|--|
| bicubic sinc | region_of_interest_interpolation_on_zoom $\equiv 0$ |
| main_display_interpolation_on_zoom | $\equiv 0$ |
| bilinear | = 1 |
| pixel replication | = 2 |
| bicubic spline | = 3 |

TRANSLATION: Data may be translated fractional pixel amounts. There are three supported methods for the interpolation in fractional pixel translations which may be changed on-the-fly:

| | |
|--|-----------------------------------|
| bicubic sinc | interpolation_on_shift = 0 |
| bilinear | = 1 |
| bicubic damped sinc (Marc Buie) | = 2 |

DEZOOMING: Data may be dezoomed in the main display. When dezooming by the integral factors of 2,3,4, and 5 the user has four options for the method which may be changed on-the-fly:

| | |
|----------------|--------------------------------|
| mean | main_display_dezoom = 0 |
| median | = 1 |
| maximum | = 2 |
| minimum | = 3 |

FLUX CONSERVATION: When resampling images (zooming or pixel scale modification) the user may select to preserve the flux per area or preserve the total flux [zoomed image = zoomed image / zoom-factor². This parameter may be changed on-the-fly.

| | |
|-------------------------------|----------------------|
| preserve flux per area | zoom_flux = 0 |
| preserve total flux | = 1 |

PIXEL ORIGIN: The standard definition of the pixel origin (IRAF/IDL) is the pixel center, or pixel [1,1] covers the area [0.5:1.5, 0.5:1.5]. An alternate definition of the pixel origin is the lower left corner where the area of pixel [1,1] is [1:2,1:2]. In order to simplify the bookkeeping in the ROI, the ROI data are shifted forward on half of an original pixel in the display when the pixel origin is set to lower left corner. Therefore, a lower left corner pixel origin for non-zoomed data is not valid. This parameter may be changed on-the-fly.

| | |
|---|-------------------------|
| origin at the center of the pixel | pixel_origin = 0 |
| origin at the lower left corner of the pixel | = 1 |

MAIN WINDOW SIZE: The user must set the desired size of the Main Window Display. If the images to be displayed are quite large, it is advisable to scroll the display. When the **main_display_scroll** is set to 0 the **main_display_scroll_X_size** and **main_display_scroll_Y_size** are ignored. These parameters may be changed on-the-fly in the **Resize** Widget. The default values for the window sizes are:

| | |
|--|--|
| main display total image X size | main_display_default_X_size = 400 |
| main display total image Y size | main_display_default_Y_size = 400 |
| main display scroll [0=NO, 1 = YES] | main_display_scroll = 0 |
| main display scrolled image X size | main_display_scroll_X_size = 300 |
| main display scrolled image Y size | main_display_scroll_Y_size = 300 |

SHOW IMAGES WINDOW SIZE: The user may control the overall size of the Show Images Widget. Since Show Images is not a graphical widget stretching a corner of the widget will not change the effective area in the widget. The default values of the window size parameters are:

X size of Show Images
Y size of Show Images

show_images_X_size = 500
show_images_Y_size = 400

When the number of entries in the widget exceed the YSize of the widget, the widget will automatically scroll. Buttons associated with each image: Filename, [Image] On [/Off], MaskOn[/Off], Flip Y [axis], Add, etc [Image Disposition], [Show] Hdr, and CrdXY [Show Centroid Solution] cannot be scrolled. If any of these buttons fall off the right side of the widget the only solution is to increase the XSize of the widget. The X and Y size may be modified on-the-fly.

The user may control the horizontal size of the widget by selecting or deselecting to show the images path (may be changed on-the-fly) with:

show image path with filename [0=NO, 1=YES show_image_path = 0

DELETE IMAGES WINDOW SIZE: The user may control the size of the Delete Images Widget. When the number of entries exceeds the YSize, the widget will automatically scroll the display. If the XSize is insufficient to display the filename and the Delete button, the user must increase the XSize of the widget. The default values, which may be changed on-the-fly, are:

X size of Delete Images
Y size of Delete Images

del_images_X_size = 500
del_images_Y_size = 400

MASK BUILDER WINDOW SIZE: The user may control the size of the graphics display in the Image Mask Builder Widget. When images are larger than the display size the display will automatically scroll. The default values, which may be changed on-the-fly, are:

X display size for masked image
Y display size for masked image

build_mask_display_X_size = 512
build_mask_display_Y_size = 512

TEXT WINDOW SIZE: The user may control the horizontal size of the optional text window for all IDP3 messages. The default size is:

horizontal text window size **text_win_size = 80**

WIDGET PLACEMENT: The user may set the locations of most of the widgets in IDP3 by specifying the X and Y coordinates of the upper left corner of the widgets. The widgets' placement on the screen may be changed by moving them around on the screen.

WIDGET CREEP: When IDP3 widgets are closed with the “Done” or “Exit” buttons, IDP3 records the position of the widget on the screen. This allows the widgets to return to their previous positions where reinitialized. The feedback from IDL regarding the widget positions has an inherent error that varies according to the operating system. If the widget creep values are set to the default value of -1 IDP3 will apply the appropriate correction according to an internal lookup table for each supported operating system. The user may override these values by setting his own positive values for the parameters. The default values are:

widget creepage in the X direction **x_widgetcreep = -1**
widget creepage in the Y direction **y_widgetcreep = -1**

BLINK CONTROL: The blinking of images is controlled by the number of iterations, the delay between each iteration, and the delay between the blink of each frame. In order to stop the blinking before it has completed both delays must be set greater than 0. The default values, which may be modified in the Setup sub-menu of the Blink sub-menu of the Images Menu, are:

blink delay for each frame (in seconds) **blink_delay = 0.25**
blink delay for each iteration (in seconds) **iblink_delay = 0.1**
number of iterations to blink **blink_times = 10**

DIRECTORY PATHS for INPUT/OUTPUT: As images and lists of images are loaded into IDP3 and images and par files are saved from IDP3, the directory paths for these operations are saved in the parameters

Imagepath, **listpath**, **savepath**, and **parpath**, respectively. The default values are the **current working directory**. The user may initialize these parameters in the Preferences file. Otherwise, the paths are updated with each load or save operation.

FILE FILTERS: IDP3 uses the *Dialog_Pickfile* widget for selecting images, lists, and parameter files to load. The user may set a filter for each of these operations to minimize the number of files to search in the *Dialog_Pickfile* selection list. The default values, which may be changed on-the-fly, are:

| | |
|---|-----------------------------|
| filter for loading images | imfilter = '*.fit*' |
| filter for loading image lists | listfilter = '*.lst' |
| filter for loading saved parameter files | parfilter = '*.par' |

IMAGE PLANES: All images in IDP3 are 2-dimensional. Many datasets include 3-dimensional data. The user may select the default index of the third dimension to load. The input for this parameter must be a string regardless of its value. The default value, which may be modified on-the-fly, is:

| | |
|--|--------------------------|
| what planes to load in 3-D images [* = All] | load_planes = '*' |
|--|--------------------------|

IMAGE EXTENSIONS: When loading data from a multi-extension FITS file, the user may specify the name of the extension he wishes to load. IDP3 will load the first extension in the file whose EXTNAME matches the specified extname. The default value, which may be changed on-the-fly, is:

| | |
|---------------------------------------|------------------------|
| desired extension name to load | extname = 'SCI' |
|---------------------------------------|------------------------|

SAVING FITS FILES: When saving FITS images from the Main Display, the ROI, the Combine Widget, the 2-D Gauss Show, the 2-D Radial Profile, the

A Note about Widget Positions on the Display

The user may specify the positions of all of the “positionable” IDP3 widgets in the IDP3_Preferences file. These X,Y pairs specify the position of the top-left corner of the indicated widget with respect to the top-left corner of the display screen. These preferences are useful when running IDP3 on laptops or computers with small display screens. When most of the IDP3 widgets are closed (clicking on the Done button), the current position of the widget on the display screen is recorded. If the user has moved the widget on the screen it should return to the moved position when reinitialized.

A Note about Interpolation Methods

By default IDP3 uses the bi-cubic sinc interpolation for sub-pixel translations. The user may also select bi-linear or a bi-cubic damped sinc function developed by Marc Buie. For resampling images the user has the choice of bi-linear, bi-cubic sinc, or bi-cubic spline interpolation or pixel replication. Any of these four interpolation methods can be used to resample either the image in the main display or the image sub-section in the ROI display (note that each is specified in separate parameters). Different interpolation methods have advantages in different situations. The bi-cubic sinc function was implemented by RSI in C for performance but the bi-cubic spline is implemented in the IDL language and therefore is much slower.

A Note about Pixel Origin

The results of resampling data varies slightly according to the Pixel Origin definition set in the Preferences file. The default pixel origin of IDL, i.e., for the first pixel [0,0] the lower left corner of the pixel is defined to be [-0.5,-0.5] and the upper right corner is [0.5, 0.5] making the pixel center [0,0]. Some users prefer to consider the range of the pixel to be [0,0] to [1,1] making its center [0.5,0.5]. If the alternate definition is chosen IDP3 will move the data in the ROI by $\frac{1}{2}$ an original pixel when resampled. The cursor readback and centroid results will then have the lower left corner of the pixel as the origin.

A Note about Flux Conservation

When an image is resampled, one may conserve total flux in an image or conserve the flux per area. To conserve total flux when resampling, the resampled pixels are divided by the zoom-factor². If the total flux is conserved in the ROI the displayed image and the ROI plotmin and plotmax are divided by the ROI-zoom-factor². The option to conserve total flux is specified in the preferences file and may be changed on-the-fly [Edit Preferences Widget].

A Note about the 'Reference Image'

At any given time only one image in the work area, the Reference Image, may be adjusted. The user chooses which image this is by using the 'Select Image' entry under the 'Image' menu on the Main Widget's Menu bar. Alternately, the Reference Image may be selected by clicking on its name in the Show Images Widget. The Reference Image is indicated in the Show Images Widget with an '*' adjacent to the image number. The name of the Reference Image is also echoed at the top of the main display.

A Note about Image Arithmetic

Image arithmetic is limited to the disposition operations for each image in the Show Images Widget. The main display image begins as a blank canvas. As images are loaded and turned ON they interact with the main display image according to their disposition. A description of the current image dispositions is given below:

Add image display = image display + current image

Subtract image display = image display – current image

Divide image display = image display / current image

Invert image display = image display + 1.0 / current image

Average image display = image display + 1/n * current image where
n = number of ON images with disposition Average

Multiply image display = image display * current image

Min image display = Minimum (image display, current image)

Pos image display = image display + current image where negative pixels are set to 0.

Neg image display = image display + current image where positive pixels are set to 0 and negative pixels set positive.

Abs image display = image display + absolute value of current image

Operations like multiply, divide, and min should not be set for the first image to be processed. When applied with an image display set to zero the results may be null or undefined.

A Note about the Order of Operations for Each Image

The display image in IDP3 is very dynamic, i.e., with each selection or modification to an image the display image is recomputed from the original data. Regardless of the order in which the user specifies modifications to the image in the Adjust Position or Show Images Widgets there is a fixed order in which the operations are performed as the display image is recomputed. The order of operations, if selected, and where they may be selected are as follows:

- | | |
|---------------------------------|------------------------------|
| 1. Image Clipping | Adjust Position |
| 2. Flipping about the Y-Axis | Show Images |
| 3. Image Padding | Adjust Position, Show Images |
| 4. Pixel Scale Modification | Adjust Position |
| 5. Resampling | Adjust Position, Show Images |
| 6. Rotation | Adjust Position |
| 7. Fractional Pixel Translation | Adjust Position |
| 8. Flux Scaling | Adjust Position |
| 9. Image Biasing | Adjust Position |
| 10. Integral Pixel Translation | Adjust Position |

The operations available in the Adjust Position widget are ordered as above from the top to the bottom of the widget.

A Note about Supported Data Formats

IDP3 can ingest FITS, PICT, TIFF, and HDF files for both data and masks. All output images, however, are written as FITS files as a single HDU or a primary header and single image extension.

A Note about Saving Images

Images may be saved to disk as FITS files and/or to the image memory of IDP3. Once the image is saved to IDP3 memory it may be saved to disk with the Save Main Display function in the main File Menu at a later time. When saving parameters for a future IDP3 session no parameters will be saved for any images that are ONLY written to IDP3 memory. This is because there is no file on disk associated with the data in memory.

A Note about Blinking Images

Blinking is controlled by the Blink sub-menu of the Images menu in the Main Display Widget. The user may set the number of iterations, the delay between the display of each image in the series, and the delay between each iteration in the Setup sub-menu. The delays may be set to 0.9 seconds. However, if the user wishes to abort the iterations before they have finished the delays must be set to a value greater than 0.0. Otherwise, the stop interrupt cannot be processed. All ON images whose disposition is ADD or SUBTRACT (the negative of the image will be displayed) will be blinked for the selected number of iterations. If STOP Blink is selected during the blinking process and the delays are greater than 0.0 then blinking will stop and the image display will return to the composite of all ON images. If many images are blinked it is important to set the PLOTMIN and PLOTMAX to valid values for the individual images being blinked. Otherwise, the individual images being blinked may appear to wash out with the scaling for the composite image.

A Note about Batch Processing

IDP3 is inherently interactive. However, there are a few provisions for processing data in a batch-like way. The Radial Profile and Photometry Widgets have the capability for processing all ON images individually, but with a single step (Stack function). Another method is the use of the parameters list. The user may save parameters of the images currently loaded, i.e., their translations, rotations, etc. into a parameter file. This file may be edited, nominally to replace the names of the files for which the parameters apply. When the parameter file is restored the new images will be processed in bulk as were the original images for which the parameter file was written.

A Note about Memory

Large images use lots of memory. Many images also use lots of memory. When a lot of memory is being used, the program may slow down. There is a larger performance hit when working with a few very large images as opposed to many smaller images using the same amount of total memory.

A Note about Image Rotation

The user must specify the rotation angle (Clockwise) and the x and y center of rotation in the Adjust Position Widget. The image is rotated about the rotation center with that point fixed in the field. Some of the image will rotate out of the field of view. If the user applies a pad to the image this can be avoided.