



SAOImage DS9 Reference Manual

DS9 is the next version of the popular SAOtng display program. It is a Tk/Tcl application which utilizes the SAOtk widget set. It also incorporates the new X Public Access (XPA) mechanism to allow external processes to access and control its data, GUI functions, and algorithms. DS9 supports the direct display of FITS images and binary tables, multiple frame buffers, region cursor manipulation, many scale algorithms and colormaps, and easy communication with external analysis tasks. It is highly configurable and extensible to meet the evolving needs of the astronomical community.

DS9 supports advanced features such as multiple frame buffers, mosaic images, tiling, blinking, geometric markers, colormap manipulation, scaling, arbitrary zoom, rotation, pan, and a variety of coordinate systems (including Image, Physical, Detector, and WCS). DS9 also supports FTP and HTTP access. The GUI for DS9 is user configurable.

<u>The Graphical User Interface</u>	1
<u>Mouse and Keyboard</u>	25
<u>Command Line Options</u>	27
<u>XPA Access Points</u>	53
<u>Regions</u>	81
<u>Color</u>	91
<u>Printing</u>	92
<u>Contours</u>	93
<u>Coordinate Grids</u>	94
<u>Analysis</u>	95
<u>File Formats</u>	110

<u>Preferences</u>	118
<u>IRAF Support</u>	125



The Graphical User Interface

DS9s GUI as been designed and implemented for ease of use. It is composed of a menu bar, information panel, panner, magnifier, button control panel, display frames, horizontal and vertical graphs, and a colorbar. The user may reconfigure the display by showing or hiding one or more of the elements. See [Preferences](#) for more details.

Menu bar

[File](#)

[Edit](#)

[Frame](#)

[Bin](#)

[Zoom](#)

[Scale](#)

[Color](#)

[Region](#)

[WCS](#)

[Analysis](#)

[Help](#)

Tear off Menus

All DS9 menus can be used as a floating tool palette. To 'tear off' a menu, select the dashed line which is the first item of each menu. At this point, the menu will become its own window, which now can be managed by the user. Multiple copies of the same menu can be created.

Information Panel

The Information panel displays information about the current data values. The information displayed is configurable via the preferences.

Panner

The Panner allows the user to view the entire frame, along with the current viewing bounding box, image cursor, and wcs cursor. To pan the current frame, click and drag the viewing bounding box.

Magnifier

The Magnifier displays a magnified view of the current mouse cursor location. The Magnifier cursor outlines the size and orientation of one pixel, given the current frame zoom and orientation.

Button Control Panel

The Button Control panel provides duplicate menu functionality for ease of use.

Frames

FITS images are displayed in Frames. Frames may be viewed one at a time, or all at once in Tile mode. Only one frame is the Current Frame, as indicated by the Blue Border, while in Tile mode. Most functions act only on the Current Frame.

Graphs

Horizontal and Vertical graphs are available to display a cut through the current frame data. While the mouse cursor is in a graph, the information panel reflects the current data values. Graphs are not available while in Tile Mode.

Colorbar

The Colorbar displays the current colormap and current bias and contrast settings.



File Menu

The File Menu is used to load, unload, print, and display information about FITS images.

About DS9

Displays Modal Dialog box containing version number and credits.

Open

Load FITS Image or FITS Binary Table. The standard file dialog is presented.

Open Other...

- **Open Array**

Load Array file. The standard file dialog is presented.

- **Open URL**

Load FITS Image or FITS Binary Table from specified URL.

- **Open RGB Image**

Load FITS RGB Image from a FITS file with multiple extensions. The current frame must be a RGB frame for this option to be enabled.

- **Open RGB Cube**

Load FITS RGB Image from one FITS file consisting of a 3D data cube in the primary header. The current frame must be a RGB frame for this option to be enabled.

- **Open RGB Array**

Load RGB Array Image from one file consisting of a 3D data cube. The current frame must be a RGB frame for this option to be enabled.

- **Open Multi Ext as Data Cube**

Load entire FITS image composed of multiple extensions as a data cube. The standard file dialog is presented.

- **Open Mosaic IRAF**

Load entire FITS Mosaic image composed of multiple extensions. The IRAF keywords DETSIZE and DETSEC are used to construct the mosaic. The standard file dialog is presented.

- **Open Mosaic IRAF Segment**

Load one FITS Mosaic Segment. The IRAF keywords DETSIZE and DETSEC are used to construct the mosaic. The standard file dialog is presented.

- **Open Mosaic WCS**

Load entire FITS Mosaic image composed of multiple extensions. WCS keywords are used to construct the mosaic. The standard file dialog is presented.

- **Open Mosaic WCS Next**

Load an additional entire FITS Mosaic image composed of multiple extensions. A valid WCS mosaic must be already loaded. WCS keywords are used to construct the mosaic. The standard file dialog is presented.

- **Open Mosaic WCS Segment**

Load one FITS Mosaic Segment. WCS keywords are used to construct the mosaic. The standard file dialog is presented.

- **Open Mosaic WFPC2**

Load HST WFPC2 FITS Mosaic image. The standard file dialog is presented.

Preserve During Load

- **Scale**

When a new image is load into the frame, existing scale parameters are preserved.

- **Pan**

Preserve existing pan location when a new image is loaded into the frame.

- **Regions**

All existing regions are preserved when a new image is loaded.

Save as...

- **Image**

Save visible image, with regions, contours and coordinate grids as raster image.

- **Export**

Export all image data from current frame without graphics.

Display Fits Header

Displays the FITS header. If FITS Mosaic is loaded, a list of headers will be presented.. If FITS Bin Table is loaded, the table header will be displayed, not the binned image.

XPA Information

Use this menu item to display internal xpa parameters.

Source TCL

Sources a valid TCL file.

Print

Controls Postscript printing. Destination, PS Level, Color model, and Resolution can be configured.

Page Setup

Controls Postscript page layout. Orientation and page size can be configured.

Exit

Quits DS9.



Edit Menu

Undo

Allows the user to undo a move, edit or delete action on a region or group of regions.

Cut

Allows the user to *Cut* region(s) for later *Paste* within the same frame or another frame.

Copy

Allows the user to *Copy* region(s) for later *Paste* within the same frame or another frame.

Paste

If a region or group of regions have been *Cut* or *Copy*, this will recreate the regions within the same frame or another frame.

Mouse Button Modes

DS9 supports a number of modes for configuring the behaviour of the mouse when moved, clicked, or dragged over various sections of the DS9 display.

Disabled Mode

The Mouse Button 1 is disabled.

Pointer Mode

Information Panel

While in Pointer Mode, the information panel displays information concerning the data values under the current mouse cursor location. If the mouse cursor is not over a valid data value, the information panel will be blank.

Graphs

If the horizontal or vertical graphs are visible, the data displayed will be based on the current mouse cursor location. Again, if the mouse cursor is not over a valid data value, the graphs will show no data.

Arrow Keys

The arrow keys will move the mouse cursor and all selected regions, one pixel at a time, in the specified direction.

Mouse Button 1

Clicking the first mouse button allows the user to select the current frame and select, create, edit, move, and rotate regions.

- Selecting the current frame
If DS9 is in Tile Mode, if you click on a frame that is not the current frame, it becomes the current frame. Use may also use the `TAB` key to cycle through all frames.
- Deselecting regions
For the current frame, if one or more regions are currently selected, clicking outside of a region will deselect all regions. See [Regions](#) for more information.
- Selecting regions
For the current frame, clicking on a region will select that region. `SHIFT` clicking will toggle the selection of that region to the set of currently selected regions. See [Regions](#) for more information.
- Creating regions
For the current frame, if no regions are currently selected, clicking outside of a region will create a new region. See [Regions](#) for more information.
 - If you just click and release, circle, rectangle, ellipse, point, and text region can be created in the default size. You can not create a line, ruler, polygon region in this manner.
 - If you click, drag, and release, all region types can be created.
 - After a region is created, it will not be selected.
- Moving regions
For the current frame, a region can be moved by clicking on and dragging. To move multiple regions at the same time, select the regions first, then move. If you click on a non selected region while other regions are selected, the other regions are deselected, and only the clicked region is moved. See [Regions](#) for more information.
- Editing regions
For the current frame, a region can be edited (changed in size) by clicking on one of the 4 editing handles, which become visible when a region is selected. See [Regions](#) for more information.
- Rotating regions
For the current frame, a region can be rotated by `SHIFT` clicking on one of the 4 editing handles, which become visible when an region is selected. See [Regions](#) for more information.

Crosshair Mode

While in Crosshair mode, each frame will display a set of crosshairs. To move the crosshairs, click and drag the first mouse button.

Information Panel

The information panel displays information concerning the data values under the crosshair.

Graphs

If the horizontal or vertical graphs are visible, the data displayed will be based on the current crosshair location.

Arrow Keys

The arrow keys will move the crosshair in the specified direction, one pixel at a time.

Colorbar Mode

Change the contrast and bias of the current colormap by click and dragging. This function is also available at all times via the third mouse button.

Pan Mode

Pan the current image, within the current frame, by click and dragging. This function is also available at all times via the second mouse button.

Zoom Mode

Zoom the current image, within the current frame. Clicking will zoom by a factor of 2 at the current location. SHIFT clicking will zoom out by a factor of 2 about the center.

Rotate Mode

Rotate the current image, within the current frame, by click and dragging. Movement to the left will rotate counter clock wise, movement to the right will rotate clock wise. Note: this function is compute intensive, so the refresh rate can be very slow, depending on computer power.

Examine Mode

Creates a new frame and loads the base image into that frame at an increased scale factor in the same manner as the magnifier. The actual data is not re-loaded, but shared between the frames. Therefore, it is fast and requires very little additional memory. The user may indicate via the Preferences, if a new frame is created each time or the same frame is used. If the base frame is cleared or deleted, all examine frames will be deleted. It is possible to 'examine' a frame that was created by examine.

Preferences

Allows the user to customize the appearance and behavior of the GUI . See [Preferences](#) for more information on using and saving preferences.



Frame Menu

New Frame

Creates a new frame. Each frame has a unique name.

Delete Frame

Unloads any FITS files and deletes the current frame.

Delete All Frames

Unloads all FITS files and deletes all frames.

Clear Frame

Clears or Unloads any FITS files in the frame.

Reset Frame

Resets the current frame to default settings.

Refresh Frame

This allows the user to refresh or rerender the current frame. This is useful for real-time applications where the underlining shared memory segment or mmap file has changed contents.

Single Frame

Sets the display mode to single frame. One frame is displayed at a time. You can use the TAB key or the Frame Menu to advance to the next available frame. Only frames that are not 'Hidden' are available.

Tile Frames

Sets the display mode to multiple frame mode. When in TILE mode, the horizontal and vertical graphs are not available. The TAB key or Frame Menu can be used to advance to the next available frame. Only frames that are not 'Hidden' will be shown.

Blink Frames

Sets the display mode to blink mode. Only frames that are not 'Hidden' will be shown.

Tile

Select the current Tile Mode.

- Grid
All frames are displayed as a grid of equal number of rows and columns.
- Column
All frames are displayed in a number of columns.
- Row
All frames are displayed in a number of rows.

Blink Interval

Set the delay in seconds between blinking frames or between slices of a 3D data cube.

Match Frames

Scale, Rotate, and Pan all frames referenced to the current frame based on the selected coordinate system.

Match Colorbars

Match the Colormap, Bias, and Contrast for all frames referenced to the current frame.

Match Scales

Match all scale parameters, including type, mode, scope, minmax parameters, and zscale parameters across all frames to the current frame.

Lock Crosshairs

When in crosshair mode, position the crosshair cursor for all frame referenced to the current frame based on the selected coordinate system.

Data Cube

Displays Fits Data Cube dialog.

Show/Hide Frames

Allows the user to indicate which frames are 'Hidden' or not available. When a frame is 'Hidden', it is not available for display, and can not be selected from the Frame Menu.



Bin Menu

To create an image from a FITS Bin Table, the user needs to specify a binning factor, binning buffer size, and the binning function.

By default, DS9 will bin about the center of the image. To determine the center of the image, DS9 will look for the following keywords in order:

- TDMIN/TDMAX
- TLMIN/TLMAX
- TALEN
- AXLEN

If no valid keywords are found, DS9 will define the center as the middle of the possible data space based on the coordinate data type.

Bin Function

- Average-- all pixel values that fall into one pixel bin are averaged.
- Sum-- all pixel values that fall into one pixel bin are summed.

Bin to Fit Frame

Bin to Fit Frame will calculate a bin factor as a power of 2 that will allow the entire data space to be displayed in the current frame.

Bin Block Factor

A value greater or equal to zero. This value indicates the number of pixel values that will fall into a particular bin.

Bin Buffer Size

The overall size of the image generated. This has no relation to min and max values of the columns used to create the image. The image generated is of BIN BUFFER SIZE, centered at the current view center.

Bin Smooth

Smooth image after binning. Smoothing parameters may be modified via the Binning Parameters Menu item.

Binning Parameters

Allows the user to specify an arbitrary binning parameters, including filters and 3D binning.



Zoom Menu

Use the Zoom Menu to specify the current zoom factor, orientation, and angle of the image in the current frame.

Align to WCS

If a valid WCS is present, rotate and orient the image so that the WCS is correctly displayed.

Zoom In

Zoom in the image by a factor of 2.

Zoom Out

Zoom out the image by a factor of 2.

Zoom to Fit Frame

Zoom to Fit Frame will calculate the correct zoom factor to allow the entire data space to be displayed in the current frame.

Center Frame

Will pan to the center of the current frame. Please Note:

- For FITS Bin Tables-- The center is defined by the `TLMAX`, `TLMIN`, `TALen`, or `AXLEN` FITS keywords if present. If not available, by the min and max of the data type.
- For FITS Mosaics-- The center is defined by the `DETSIZE` FITS keyword.

None

The image axes are not inverted.

Invert X

Invert the X axis of the image of the current frame.

Invert Y

Invert the Y axis of the image of the current frame.

Invert XY

Invert both the X and Y axis of the image of the current frame.

Pan Zoom Rotate To

Invoke a dialog box to specify a view center, zoom factor, and rotation angle for the current frame.



Scale Menu

The Scale Menu is used to set the upper and lower limits and the distribution of colors, based on pixel values.

Linear

Set the colors distribution to Linear.

Log

Set the colors distribution to Log.

Squared

Set the colors distribution to Squared.

Square Root

Set the colors distribution to Square Root.

Histogram Equalization

Set the colors distribution based on a Histogram Equalization algorithm.

MinMax

Set the upper and lower limits to the min and max values, or 100%. The image may be scanned to determine these values, based on the Min Max Options.

99.5% to 90%

Set the upper and lower limits to be based on the specified percentage. A histogram of the data is created and the limits are set to display the percentage, about the mean value.

ZScale

Set the upper and lower limits based on the IRAF ZScale algorithm. ZScale parameters may be configured via the preference menu.

ZMax

This is mix of MinMax, or 100% and ZScale. Set the lower limits base on the IRAF ZScale algorithm and the upper limit to the max data value.

Scope

For FITS Mosaics and Datacubes, scope specifies if the limits are applied global across each FITS Mosaic segment, or on a per segment basis.

MinMax

Specify how to determine the minium and maxium values of the image. A selection of AUTO will use SAMPLE for all FITS Mosaics and Data Cubes, and SCAN for all other files.

use DATASEC

If the keyword DATASEC is present, and this option is enabled, data not in the bounding box specified by DATASEC will not be imaged, printed, or used in calculations of MinMax and ZScale. The excluded regions are usually bias strips on CCDs and contain no real data.

Scale Parameters

Invoke a non-modal dialog box to specify Scale Type, Scale Limits, and Scale parameters.



Color Menu

Colormaps

Select the colormap for the current frame. All loaded colormaps are listed.

Invert Colormap

Toggle to invert current colormap.

Reset Colormap

Reset current colormap to default contrast/bias.

Colormap Parameters

Displays Colormap Dialog. Users can Load/Save Colormaps, Load/Save Contrast/Bias settings, and specify current contrast /bais parameters.



Region Menu

The Region Menu is used to create, configure, and delete regions in the current frame. For more information on supported region types, properties, colors, fonts, and regions file formats, See [Regions](#).

Show Regions

Toggles the visibility of regions.

Get Info

Will display one dialog box for each region that is selected in the current frame. The type of dialog box differs with the type of the region displayed.

Shape

Select the region shape for all new regions. This does not change the current shape of selected regions.

Color

Select the color of new regions or selected regions in the current frame

Width

Select the width of new regions or selected regions in the current frame.

Properties

Select the properties of new regions or selected regions in the current frame.

Font

Select the font of new regions or selected regions in the current frame.

Move to Front

Move all selected regions in the current frame, forward in the display list.

Move to Back

Move all selected regions in the current frame, back in the display list. Then the regions will no longer be selected.

Select All

Select all regions in the current frame.

Select None

Unselect all regions in the current frame.

Delete Selected Regions

Delete all regions that are selected in the current frame.

Delete All Regions

Delete all regions in the current frame.

New Group

Creates a new regions group from selected regions

Groups

Opens the groups dialog box in which groups can be edited and deleted.

List Regions

Lists all regions in the current frame in the format and coordinate system as indicated in the Format and File Coordinate System menu items.

Load Regions

Load a regions file into the current frame.

Save Regions

Save as a regions file all regions in the current frame. The regions format and coordinate systems used is indicated via the Format and File Coordinate System menu items.

File Format

Specify the regions file format. Supported formats are DS9 / Ftools, Ciao, SAOtngr, SAOimage, IRAF pro and X Y coordinate pairs. See [Regions](#) for more information.

Strip Output

Stripped output removes all comments and properties, places all regions on one line, and delimits each line with a semicolon.

File Coordinate System

Specify the coordinate system for input and output. Not all coordinate systems are supported by all regions formats. Furthermore, most regions formats ignore this for input.



WCS Menu

The WCS Menu is used to change the current WCS parameters for a number of functions. When one of the WCS parameters is modified using this menu, the following gui features are modified to reflect the new setting. This menu is provided as a feature to enable the user to quickly change and synchronize the wcs settings for a number of related features.

- Information Panel
- Align to WCS
- Panner WCS Compass
- Grid
- Regions File Coordinate System
- Print Coordinates
- Pan Rotate Zoom Dialog Box



Analysis Menu

The Analysis Menu contains items used in the analysis of image data. In addition to the menu items listed below, other items may be added by the user via TCL scripts.

Horizontal Graph

Toggle the display of the horizontal graph. If in tile mode, the horizontal graph is not available.

Vertical Graph

Toggle the display of the vertical graph. If in tile mode, the vertical graph is not available.

Display Contours

Toggle the display of contours using the current contour parameters.

Contours Parameters

Create, copy, paste, and configure contours from image data. For more information, see [Contours](#).

Display Coordinate Grid

Toggle the display of a coordinate grid using the current grid parameters.

Coordinate Grid Parameters

Display a non-modal dialog box to allow modification of coordinate grid parameters. For more information, see [Coordinate Grids](#).

Name Resolution

Enables name resolution using NED or SIMPBAD. Internet access must be available for this feature to work.

DSS Server

Enables access to the DSS image servers at SAO, ESO, and STSI. Internet access must be available for this feature to work.

Archive Servers

Brings up the internal DS9 Web Display tool and load the specified URL. Based on the site selected, certain web form fields will be initialized based on the image displayed in the current frame.

Virtual Observatory

Enables access to supported virtual observatories from DS9. Internet access must be available for this feature to work.

Plot Tool

Bring up a blank Plot tool window. Allows the user to load/save/plot 2D and 3D data quickly.

Web Display

Bring up the internal DS9 Web Display tool.

Analysis Command Log

Display the complete command line before execution of an analysis command

Load Analysis Commands

Load a new set of Analysis commands and add to the existing analysis menus.

Clear Analysis Commands

Clear all analysis commands and menus.



Help Menu

The Help Menu is used for accessing all documentation for DS9. New to version 2.2, DS9 documentation is in HTML format and is stored internally. No external web browser, nor internet access is need to access the documentation.

Reference Manual

Display the Reference Manual.

Keyboard Short Cuts

Display all keyboard and mouse bindings

FAQ

Frequently Ask Questions. Display a short discussion on common questions.

New Features

A short discussion on new features that have been added recently.

Known Issues

A list of know issues, bugs, and *features*.

Release Notes

Latest release notes for this version of ds9.

Help Desk

Contact information for detailing any problems, questions, or suggestions you may have concerning the use of DS9.

DS9 Home Page

Displays the DS9 Home Page in the interal web browser. Can be used to download the latest DS9 version.



Mouse and Keyboard

Mouse Buttons

The following table contains the event bindings for the mouse buttons.

Mouse Button	Description
Button 1	Depends on current MODE, which may be selected by the <u>EDIT</u> menu option.
Button 2	Pan the current image, within the current frame. Behavior depends on the PAN preference settings.
Button 3	Change the contrast and bias of the current colormap by click and dragging.

Keyboard Shortcuts

The following table contains the list of keyboard shortcuts and the resulting action taken.

Key Stroke	Description
TAB	Goto next frame
Shift-TAB	Goto previous frame
DELETE	Deletes selected regions
c	Print Mouse Coordinates and Pixel value.
f	Toggles Infobox freeze
i	Set include property for region
e	Set exclude property for region
s	Set source property for region
b	Set background property for region
g	Create a new group
Shift-g	Create a new group with default name
+	Goto next 3D Fits Slice
-	Goto previous 3D Fits Slice
Up Arrow k	Will move selected regions up one pixel. In Pointer mode, will move the cursor up one pixel. In Crosshair mode, will move the crosshair up one pixel. In Pan mode, will pan the image up one pixel.
Right Arrow l	Will move selected regions to the right one pixel. In Pointer mode, will move the cursor to the right one pixel. In Crosshair mode, will move the crosshair to the right one pixel. In Pan mode, will pan the image to the right one pixel.
Left Arrow h	Will move selected regions to the left one pixel. In Pointer mode, will move the cursor to the left one pixel. In Crosshair mode, will move the crosshair to the left one pixel. In Pan mode, will pan the image to the left one pixel.
Down Arrow j	Will move selected regions down one pixel. In Pointer mode, will move the cursor up down one pixel. In Crosshair mode, will also move the crosshair down one pixel. In Pan mode, will pan the image down one pixel.
Shift-Drag	Will select all regions within the indicated region
Control-Drag	On selected ANNULUS Regions, will create new radii



Command Line Options

DS9 will process each command line option, one at a time, as the last step in the initialization process. Therefore, it is possible to use command line options as a little script. For example, the following command line option is used:

```
$ds9 -tile foo.fits -cmap Heat -zscale bar.fits -cmap I8
```

First DS9 is put in tile mode, then `foo.fits` is loaded. Then the colormap for `foo.fits` is changed to `Heat`. Next, a new frame is created and `bar.fits` is loaded. And finally, the colormap for `bar.fits` is changed to `I8`.

- about
- analysis
- array
- bin
- blink
- blue
- cmap
- contour
- datacube
- dss
- exit
- fifo
- fifo_only
- fits
- sfits
- frame
- geometry
- green
- grid
- help
- histegu
- iconify
- inet_only
- invert
- linear
- lock
- log

- lower
- match
- medatacube
- minmax
- mode
- mosaicimage
- mosaicimagenext
- mosaic
- smosaic
- nameserver
- orient
- pagesetup
- pan
- pixeltable
- plot
- prefs
- preserve
- print
- private
- port
- port_only
- quit
- raise
- region
- rgb
- rgbarray
- rgbcube
- srgbcube
- rgbimage
- red
- rotate
- saveimage
- savefits
- savempeg
- scale
- shm
- single
- squared
- sqrt
- source
- tile
- title
- unix

- unix_only
- url
- version
- view
- visual
- vo
- wcs
- web
- xpa
- zmax
- zscale
- zoom

about

Get DS9 credits.

Syntax:

-about

Example:

```
$ds9 -about
```

analysis

Load analysis file.

Syntax:

-analysis filename

Example:

```
$ds9 -analysis my.analysis
```

array

Load an array.

Syntax:

-array

```
<filename>[[xdim=<x>,ydim=<y>|dim=<dim>],zdim=<z>,bitpix=<b>,skip=<s>,
    arch=[littleendian|bigendian]]
```

Example:

```
$ds9 -array bar.arr[xdim=512,ydim=512,zdim=1,bitpix=16] # load
512x512 short
```

```
$ds9 -array bar.arr[dim=256,bitpix=-32,skip=4] # load 256x256 float
with 4 byte head
```

```
$ds9 -array bar.arr[dim=512,bitpix=32,arch=littleendian] # load
512x512 long, intel
```

bin

Controls binning factor, binning buffer size, and binning function for binning FITS bin tables.

Syntax:

```
-bin [about <x> <y>]
      [buffer size <value>]
      [cols <x> <y>]
      [factor <value> [<value>]]
      [depth <value>]
      [filter <string>]
      [function average|sum]
      [to fit]
      [smooth yesno]
      [smooth function boxcar|tophat|gaussian]
      [smooth radius <value>]
```

Example:

```
$ds9 -bin about 4096 4096
$ds9 -bin buffer size 512
$ds9 -bin cols detx dety
$ds9 -bin factor 4
$ds9 -bin factor 4 2
$ds9 -bin depth 10
$ds9 -bin filter '{pha > 5}'
$ds9 -bin function sum
$ds9 -bin to fit
$ds9 -bin smooth yes
$ds9 -bin smooth function boxcar
$ds9 -bin smooth radius 3
```

blink

Activates Blink Mode.

Syntax:

```
-blink
```

Example:

```
$ds9 -blink
```

blue

For RGB frames, sets the current color channel to blue.

Syntax:

```
-blue
```

Example:

```
$ds9 -blue foo.fits
```

cmap

Controls the colormap for the current frame. The colormap name is not case sensitive. A valid contrast value is from 0 to 10 and bias value from 0 to 1.

Syntax:

```
-cmap [<colormap>]
      [file <filename>]
      [invert yes|no]
      [value <contrast> <bias>]
```

Example:

```
$ds9 -cmap Heat
$ds9 -cmap file foo.sao
$ds9 -cmap invert yes
$ds9 -cmap value 5 .5
```

contour

Controls contours in the current frame.

Syntax:

```
-contour [yes|no]
        [copy]
        [clear]
        [paste <coordinate system> [<sky frame>] <color> <width>]
        [load <filename> <coordinate system> [<sky frame>] <color>
<width>]
        [save <filename> <coordinate system> [<sky frame>]]
```

Example:

```
$ds9 -contour
$ds9 -contour yes
$ds9 -contour copy
$ds9 -contour clear
$ds9 -contour paste wcs fk4 red 2
$ds9 -contour load foo.con wcs fk5 yellow 2
$ds9 -contour save foo.con wcs fk5
```

datacube

Controls FITS datacube.

Syntax:

```
-datacube [play|stop|next|prev|first|last]
          [#]
          [interval #]
```

Example:

```
$ds9 -datacube play
$ds9 -datacube last
$ds9 -datacube 3
$ds9 -datacube interval 2
```

dss

Support for Digital Sky Survey. The first three options will download an image based on the name or specified fk5 coordinate. The remaining options are used to configure downloads.

Syntax:

```
-dss [<object>]
      [name <object>]
      [coord <ra> <dec>] # in wcs fk5

      [server sao|stsci|eso]
      [survey dss|dss2red|dss2blue]
      [size <width> <height>] # in arcmin
```

Example:

```
$ds9 -dss m31
$ds9 -dss name m31
$ds9 -dss coord 00:42:44.404 +41:16:08.78
$ds9 -dss server eso
$ds9 -dss survey dss2red
$ds9 -dss size 10 10
```

exit

quit

Quits DS9.

Syntax:

```
-exit
-quit
```

Example:

```
$ds9 -exit
```

fifo

Set the name of the IRAF input and output fifos. The default is /dev/imt1. These fifos are used by IRAF to communicate with DS9.

Syntax:

```
-fifo name
```

Example:

```
$ds9 -fifo /dev/imt1
```

fifo_only

Only use IRAF input and output fifos. Same as -port 0 -unix none.

Syntax:

```
-fifo_only
```

Example:

```
$ds9 -fifo_only
```

fits

Load FITS file.

Syntax:

```
-fits <filename>
```

Example:

```
$ds9 -fits foo.fits
```

sfits

Load split FITS file.

Syntax:

```
-sfits <filename> <filename>
```

Example:

```
$ds9 -sfits foo.hdr foo.arr
```

frame

Controls frame functions. Frames may be created, deleted, reset, and centered. While return the current frame number. If you goto a frame that does not exists, it will be created. If the frame is hidden, it will be shown. The 'frameno' option is available for backward compatibility.

Syntax:

```
-frame [center [#|all]]  
      [clear [#|all]]  
      [new [rgb]]  
      [delete [#|all]]  
      [reset [#|all]]  
      [refresh [#|all]]  
      [hide [#|all]]  
      [show [#|all]]  
      [first]  
      [next]  
      [prev]  
      [last]  
      [frameno #]  
      [#]
```

Example:

```
$ds9 -frame center  
$ds9 -frame center 1  
$ds9 -frame center all  
$ds9 -frame clear  
$ds9 -frame new  
$ds9 -frame new rgb  
$ds9 -frame delete  
$ds9 -frame reset
```

```
$ds9 -frame refresh
$ds9 -frame hide
$ds9 -frame show 1
$ds9 -frame first
$ds9 -frame next
$ds9 -frame prev
$ds9 -frame last
$ds9 -frame frameno 4
$ds9 -frame 3
```

geometry

Define the initial window geometry. see X(1).

Syntax:

```
-geometry value
```

Example:

```
$ds9 -geometry 640x480
```

green

For RGB frames, sets the current color channel to green.

Syntax:

```
-green
```

Example:

```
$ds9 -green foo.fits
```

grid

Controls coordinate grid.

Syntax:

```
-grid [yes|no]
      [load <filename>]
      [save <filename>]
      [system <coordinate system>]
      [sky <sky frame>]
      [skyformat <skyformat>]
      [type analysis|publication]
      [type axes interior|exterior]
      [type numerics interior|exterior]
      [view grid|axes|title|border|vertical yes|no]
      [view axes numbers|tickmarks|label yes|no]
```

Example:

```
$ds9 -grid
$ds9 -grid yes
$ds9 -grid load foo.grd
$ds9 -grid save foo.grd
$ds9 -grid system wcs
```

```
$ds9 -grid sky fk5
$ds9 -grid skyformat degrees
$ds9 -grid type analysis
$ds9 -grid type axes interior
$ds9 -grid type numerics interior
$ds9 -grid view grid yes
$ds9 -grid view axes yes
$ds9 -grid view axes numbers yes
$ds9 -grid view axes tickmarks yes
$ds9 -grid view axes label yes
$ds9 -grid view title yes
$ds9 -grid view border yes
$ds9 -grid view vertical no
```

help

Displays list of Command Line Options. Users of *cs*h and *tc*sh may have to quote.

Syntax:

```
-?
```

```
-- -help
```

Example:

```
$ds9 -- -help
```

```
$ds9 '-?'
```

histequ

Select histogram equalization scale for current frame.

Syntax:

```
-histequ
```

Example:

```
$ds9 -histequ
```

iconify

DS9 is iconified at startup.

Syntax:

```
-iconify
```

Example:

```
$ds9 -iconify
```

invert

Invert Colormap.

Syntax:

```
-invert
```

Example:

```
$ds9 -invert
```

linear

Select Linear scale for current frame.

Syntax:

```
-linear
```

Example:

```
$ds9 -linear
```

lock

Lock frames.

Syntax:

```
lock [crosshairs none|wcs|wcsa...wcsz|physical|image]
```

Example:

```
$ds9 -lock crosshairs wcs
```

log

Select LOG scale for current frame. This is the same algorithm as used in *IRAF*.

Syntax:

```
-log
```

Example:

```
$ds9 -log
```

lower

Lower ds9 in the window stacking order.

Syntax:

```
-lower
```

Example:

```
$ds9 -lower
```

match

Match all other frames to the current frame.

Syntax:

```
-match [frames wcs|physical|image]  
      [colorbars]  
      [scales]
```

Example:

```
$ds9 -match frames wcs
```

```
$ds9 -match colorbars
```

```
$ds9 -match scales
```

medatacube

Load multiple extension FITS file as a data cube.

Syntax:

```
-medatacube <filename>
```

Example:

```
$ds9 -medatacube foo.fits
```

minmax

This is how DS9 determines the min and max data values from the data. SCAN will scan all data. SAMPLE will sample the data every n samples. DATAMIN and IRAFMIN will use the values of the keywords if present. In general, it is recommended to use SCAN unless your computer is slow or your data files are very large. Select the increment interval for determining the min and max data values during sampling. The larger the interval, the quicker the process.

Syntax:

```
-minmax [ ]  
        [scan|sample|datamin|irafmin]  
        [mode scan|sample|datamin|irafmin]  
        [interval <value>]
```

Example:

```
$ds9 -minmax  
$ds9 -minmax scan  
$ds9 -minmax mode sample  
$ds9 -minmax interval 10
```

mode

Select the current mode.

Syntax:

```
-mode [none|pointer|crosshair|colorbar|pan|zoom|rotate]
```

Example:

```
$ds9 -mode crosshair
```

mosaicimage

Load entire FITS Mosaic image composed of multiple extensions in one FITS file. All extensions that are FITS Images and are valid FITS Mosaic Images will be loaded.

Syntax:

```
-mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] <filename>
```

Example:

```
$ds9 -mosaicimage iraf bar.fits  
$ds9 -mosaicimage wcs bar.fits  
$ds9 -mosaicimage wcsa bar.fits  
$ds9 -mosaicimage wfpc2 hst.fits
```

mosaicimagenext

Load entire FITS WCS Mosaic image composed of multiple extensions in one FITS file with out clearing the current frame. All extensions that are FITS Images and are valid FITS Mosaic Images will be loaded.

Syntax:

```
-mosaicimagenext [iraf|wcs|wcsa...wcsz|wfpc2] <filename>
```

Example:

```
$ds9 -mosaicimagenext wcs bar.fits
```

mosaic

Load FITS Mosaic image segment.

Syntax:

```
-mosaic [iraf|wcs|wcsa...wcsz] <filename>
```

Example:

```
$ds9 -mosaic iraf foo.fits
```

```
$ds9 -mosaic wcs bar.fits
```

smosaic

Load split FITS Mosaic image segment.

Syntax:

```
-smosaic [iraf|wcs|wcsa...wcsz] <filename><filename>
```

Example:

```
$ds9 -smosaic iraf foo.hdr foo.arr
```

```
$ds9 -smosaic wcs foo.hdr foo.arr
```

nameserver

Support Name Server functions. Coordinates are in fk5.

Syntax:

```
-nameserver [<object>]
             [name <object>]
             [server ned-sao|ned-eso|simbad-sao|simbad-eso]
             [skyformat degrees|sexagesimal]
```

Example:

```
$ds9 -nameserver m31
```

```
$ds9 -nameserver name m31
```

```
$ds9 -nameserver server ned-sao
```

```
$ds9 -nameserver skyformat sexagesimal
```

orient

Controls the orientation of the current frame.

Syntax:

```
-orient [none|x|y|xy]
```

Example:

```
$ds9 -orient xy
```

pagesetup

Controls Page Setup options.

Syntax:

```
-pagesetup [orientation portrait|landscape]
           [pagescale scaled|fixed]
           [pagesize letter|legal|tabloid|poster|a4]
```

Example:

```
$ds9 -pagesetup orientation portrait
```

```
$ds9 -pagesetup pagescale scaled
```

```
$ds9 -pagesetup pagesize poster
```

pan

Controls the current image cursor location for the current frame.

Syntax:

```
-pan [x y <coordinate system> [<sky frame>][<sky format>]
     [to x y <coordinate system> [<sky frame>][<sky format>]]
```

Example:

```
$ds9 -pan 200 200 image
```

```
$ds9 -pan to 400 400 physical
```

```
$ds9 -pan to 13:29:55 47:11:50 wcs fk5
```

pixeltable

nopixeltable

Display/Hide pixel table

Syntax:

```
-pixeltable
```

```
-nopixeltable
```

Example:

```
$ds9 -pixeltable
```

plot

Display and configure data plots. All plot commands take an optional second command, the plot name. Use `xpaget plot` to retrieve all plot names. If no plot name is specified, the last plot created is assumed. Plot data is assumed to be a pair of coordinates, with optional error values. The follow are valid data descriptions:

xy	x and y coordinates
xyex	x,y coordinates with x errors
xyey	x,y coordinates with y errors
xyexey	x,y coordinates with both x and y errors

To create a new plot, use the `plot new` command. If the second arg is `stdin`, the title, x axis title, y axis title, and dimension are assumed to be on the first line of the data.

Syntax:

```
# create new empty plot window
-plot []
    [new [name <plotname>]]
# edit existing plot
-plot [<plotname>] [close]
    [<plotname>] [clear]
    [<plotname>] [load <filename> xy|xyex|xyey|xyexey]
    [<plotname>] [save <filename>]
    [<plotname>] [loadconfig <filename>]
    [<plotname>] [saveconfig <filename>]
    [<plotname>] [print]
    [<plotname>] [printdestination printer|file]
    [<plotname>] [print command <command>]
    [<plotname>] [print filename <filename>]
    [<plotname>] [print palette color|gray|mono]
    [<plotname>] [page orientation portrait|landscape]
    [<plotname>] [page pagescale scaled|fixed]
    [<plotname>] [page pagesize letter|legal|tabloid|poster|a4]
    [<plotname>] [graph grid yes|no]
    [<plotname>] [graph scale
linear|linear|linearlog|loglinear|loglog]
    [<plotname>] [graph range x|y auto yes|no]
    [<plotname>] [graph range x|y min <value>]
    [<plotname>] [graph range x|y max <value>]
    [<plotname>] [graph labels title|xaxis|yaxis <value>]
    [<plotname>] [font numbers|labels|title font
times|helvetica|symbol|courier]
    [<plotname>] [font numbers|labels|title size <value>]
    [<plotname>] [font numbers|labels|title style
plain|bold|italic]
# edit current dataset
-plot [<plotname>] [dataset #]
    [<plotname>] [view discrete|line|step|quadratic|errorbar
yes|no]
    [<plotname>] [color
discrete|line|step|quadratic|errorbar<color>]
    [<plotname>] [line discrete circle|diamond|plus|cross]
    [<plotname>] [line line|step|quadratic|errorbar width <value>]
    [<plotname>] [line line|step|quadratic dash yes|no]
```

```
[<plotname>] [line errorbar style 1|2]
```

Example:

```
# create new empty plot window
$ds9 -plot
$ds9 -plot new
$ds9 -plot new name foo
# edit existing plot
$ds9 -plot close # close last plot
$ds9 -plot foo close # close plot foo
$ds9 -plot clear # clear all datasets
$ds9 -plot load foo.dat xy # load new dataset with dimension xy
$ds9 -plot save bar.dat # save current dataset
$ds9 -plot loadconfig foo.plt # load plot configuration
$ds9 -plot saveconfig bar.plt # save current plot configuration
$ds9 -plot print
$ds9 -plot print destination file
$ds9 -plot print command "lp"
$ds9 -plot print filename "foo.ps"
$ds9 -plot print palette gray
$ds9 -plot page orientation portrait
$ds9 -plot page pagescale scaled
$ds9 -plot page pagesize letter
$ds9 -plot graph grid yes
$ds9 -plot graph scale loglog
$ds9 -plot graph range x auto yes
$ds9 -plot graph range x min 0
$ds9 -plot graph range x max 100
$ds9 -plot graph range y auto yes
$ds9 -plot graph range y min 0
$ds9 -plot graph range y max 100
$ds9 -plot graph labels title {The Title}
$ds9 -plot graph labels xaxis {X}
$ds9 -plot graph labels yaxis {Y}
$ds9 -plot font numbers font times
$ds9 -plot font numbers size 12
$ds9 -plot font numbers style bold
$ds9 -plot font labels font times
$ds9 -plot font title font times
# edit current dataset
$ds9 -plot dataset 2 # set current dataset to the second dataset
loaded
$ds9 -plot view discrete yes
$ds9 -plot color discrete red
$ds9 -plot line discrete cross
$ds9 -plot line step width 2
$ds9 -plot line step dash yes
$ds9 plot line errorbar style 2
```

port

Set the IRAF port number, used by IRAF to communicate with DS9. The default is 5137, the standard IRAF port used by *ximtool*.

Syntax:

```
-port number
```

Example:

```
$ds9 -port 5137
```

port_only

inet_only

Only use the IRAF port number. This is the same as `-fifo none -unix none`.

Syntax:

```
-port_only
```

Example:

```
$ds9 -port_only
```

prefs

Controls various preference settings.

Syntax:

```
prefs [mosaicfast yes|no]
      [bgcolor white|black|red|green|blue|cyan|magenta|yellow]
      [nancolor white|black|red|green|blue|cyan|magenta|yellow]
```

Example:

```
$ds9 -prefs mosaicfast no
$ds9 -prefs bgcolor black
$ds9 -prefs nancolor red
```

preserve

Preserve the follow attributes while loading a new image.

Syntax:

```
preserve [scale yes|no]
         [pan yes|no]
         [regions yes|no]
```

Example:

```
$ds9 -preserve scale yes
$ds9 -preserve pan yes
$ds9 -preserve regions yes
```

print

Controls printing. Use `print` option to set printing options. Use `print` to actually print.

Syntax:

```
-print [destination printer|file]
       [command <command>]
       [filename <filename>]
       [palette rgb|cmyk|gray]
       [level 1|2]
       [interpolate yes|no]
       [resolution 53|72|75|150|300|600]
```

Example:

```
$ds9 -print
$ds9 -print destination file
$ds9 -print command 'gv -'
$ds9 -print filename foo.ps
$ds9 -print palette cmyk
$ds9 -print level 2
$ds9 -print interpolate no
$ds9 -print resolution 75
```

private

use private colormap, valid for pseudocolor 8 mode.

Syntax:

```
-private
```

Example:

```
$ds9 -private
```

raise

Raise ds9 in the window stacking order.

Syntax:

```
-raise
```

Example:

```
$ds9 -raise
```

region

Loads a Regions File into the current frame.

Syntax:

```
-region filename
```

Example:

```
$ds9 -region catalog.reg
```

red

For RGB frames, sets the current color channel to red.

Syntax:

```
-red
```

Example:

```
$ds9 -red foo.fits
```

rgb

Create RGB frame and control RGB frame parameters.

Syntax:

```
-rgb # empty
    [red|green|blue]
    [channel [red|green|blue]]
    [view [red|green|blue] [yes|no]]
    [system <coordinate system>]
    [lock scale|bin|colorbar [yes|no]]
```

Example:

```
$ds9 -rgb # create new rgb frame
$ds9 -rgb red # set current channel to red
$ds9 -rgb channel red # set current channel to red
$ds9 -rgb view blue no # turn off blue channel
$ds9 -rgb system wcs # set rgb coordinate system
$ds9 -rgb lock scale yes # lock rgb channels for scaling
$ds9 -rgb lock bin yes # lock rgb channels for binning
$ds9 -rgb lock colorbar yes # lock rgb colorbar channels
```

rgbarray

Load entire RGB image composed of a 3D Array Data Cube that contains red, green, and blue channels.

Syntax:

```
-rgbarray
<filename>[xdim=<x>,ydim=<y>|dim=<dim>,zdim=3],bitpix=<b>,[skip=<s>]]
```

Example:

```
$ds9 -rgbarray rgb.arr[dim=200,zdim=3,bitpix=-32]
```

rgbcube

Load entire RGB image composed of a FITS 3D Data Cube that contains red, green, and blue channels.

Syntax:

```
-rgbcube <filename>
```

Example:

```
$ds9 -rgbcube rgb.fits
```

srgbcube

Load entire RGB image composed of a split FITS 3D Data Cube that contains red, green, and blue channels.

Syntax:

```
-srgbcube <filename> <filename>
```

Example:

```
$ds9 -srgbcube rgb.hdr rgb.arr
```

rgbimage

Load entire RGB image composed of a FITS multiple extension file that contains red, green, and blue channels.

Syntax:

```
-rgbimage <filename>
```

Example:

```
$ds9 -rgbimage rgb.fits
```

rotate

Controls the rotation angle (in degrees) of the current frame.

Syntax:

```
-rotate [<value>]  
        [to <value>]
```

Example:

```
$ds9 -rotate 45
```

```
$ds9 -rotate to 30
```

saveimage

Save visible image(s) as a raster. If image is a data cube, the mpeg option will cycle thru each slice creating a mpeg movie.

Syntax:

```
-saveimage [fits|jpeg|tiff|png|ppm|mpeg] <filename>
```

Example:

```
$ds9 -saveimage jpeg ds9.jpg
```

savefits

Save current frame data as FITS. This differs from SAVEIMAGE in that the entire image of the current frame is saved as a FITS, without graphics.

Syntax:

```
-savefits <filename>
```

Example:

```
$ds9 -savefits ds9.fits
```

savempeg

Save all active frames as a mpeg movie.

Syntax:

```
-savempeg <filename>
```

Example:

```
$ds9 -savempeg ds9.mpg
```

scale

Controls the limits, color scale distribution, and use of DATASEC keyword.

Syntax:

```
-scale [linear|log|squared|sqrt|histequ]
      [datasec yes|no]
      [limits <minvalue> <maxvalue>]
      [mode minmax|<value>|zscale|zmax]
      [scope local|global]
```

Example:

```
$ds9 -scale datasec yes
$ds9 -scale histequ
$ds9 -scale limits 1 100
$ds9 -scale mode zscale
$ds9 -scale mode 99.5
$ds9 -scale scope local
```

shm

Load a shared memory segment into the current frame.

Syntax:

```
-shm [<key> [<filename>]]
    [key <id> [<filename>]]
    [shmid <id> [<filename>]]
    [fits [key|shmid] <id> [<filename>]]
    [sfits [key|shmid] <id> <id> [<filename>]]
    [mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] [key|shmid] <id>
[<filename>]]
    [mosaicimagenext [wcs|wcsa...wcsz] [key|shmid] <id>
[<filename>]]
    [mosaic [iraf|wcs|wcsa...wcsz] [key|shmid] <id> [<filename>]]
    [smosaic [iraf|wcs|wcsa...wcsz] [key|shmid] <id> <id>
[<filename>]]
    [rgbcube [key|shmid] <id> [<filename>]]
    [srgbcube [key|shmid] <id> <id> [<filename>]]
    [rgbimage [key|shmid] <id> [<filename>]]
    [rgbarray [key|shmid] <id>
[xdim=<x>,ydim=<y>|dim=<dim>,zdim=3],bitpix=<b>,[skip=<s>]]
    [array [key|shmid] <id>
```

```
[xdim=<x>,ydim=<y>|dim=<dim>],bitpix=<b>,[skip=<s>]]
```

Example:

```
$ds9 -shm 102
$ds9 -shm key 102
$ds9 -shm shmid 102 foo
$ds9 -shm fits 100 foo
$ds9 -shm sfits 100 101 foo
$ds9 -shm mosaicimage iraf key 100 foo
$ds9 -shm mosaicimage wcs key 100 foo
$ds9 -shm mosaicimage wcsa key 100 foo
$ds9 -shm mosaicimage wfpc2 key 100 foo
$ds9 -shm mosaicimagenext wcs key 100 foo
$ds9 -shm mosaic iraf key 100 foo
$ds9 -shm mosaic wcs key 100 foo
$ds9 -shm smosaic wcs key 100 101 foo
$ds9 -shm rgbcube key 100 foo
$ds9 -shm srgbcube key 100 101 foo
$ds9 -shm rgbimage key 100 foo
$ds9 -shm rgbarray shmid 102 [dim=32,zdim=3,bitpix=-32]
$ds9 -shm array shmid 102 [dim=32,bitpix=-32]
```

single

Set display mode to single.

Syntax:

```
-single
```

Example:

```
$ds9 -single
```

squared

Select Squared scale for current frame.

Syntax:

```
-squared
```

Example:

```
$ds9 -squared
```

sqrt

Select Square Root scale for current frame.

Syntax:

```
-sqrt
```

Example:

```
$ds9 -sqrt
```

source

Sources a valid TCL file.

Syntax:

```
-source filename
```

Example:

```
$ds9 -source extensions.tcl
```

tile

Controls the tile display mode.

Syntax:

```
-tile [yes|no]
      [mode grid|column|row]
      [grid]
      [grid mode [automatic|manual]]
      [grid layout <row> <col>]
      [grid gap <pixels>]
      [row]
      [column]
```

Example:

```
$ds9 -tile
$ds9 -tile yes
$ds9 -tile mode row
$ds9 -tile grid
$ds9 -tile grid mode manual
$ds9 -tile grid layout 5 5
$ds9 -tile grid gap 10
$ds9 -tile row
$ds9 -tile column
```

title

Changes the display window title to the specified name.

Syntax:

```
-title name
```

Example:

```
$ds9 -title Voyager
```

unix

Set the IRAF unix socket name, used by IRAF to communicate with DS9. The default is `/tmp/.IMT%d`, so that the standard IRAF unix socket is defined.

Syntax:

```
-unix name
```

Example:

```
$ds9 -unix "/tmp/.IMT%d"
```

unix_only

Only use the IRAF unix socket name. This is the same as `-fifo none -port 0`.

Syntax:

```
-unix_only
```

Example:

```
$ds9 -unix_only
```

url

Load FITS file from specified url.

Syntax:

```
-url <url>
```

Example:

```
$ds9 -url 'ftp://foo.bar.edu/img.fits'
```

version

Returns the current version of DS9.

Syntax:

```
-version
```

Example:

```
$ds9 -version
```

view

Controls the GUI and visible RGB frame color channels.

Syntax:

```
-view [info yes|no]
      [panner yes|no]
      [magnifier yes|no]
      [buttons yes|no]
      [colorbar yes|no]
      [horzgraph yes|no]
      [vertgraph yes|no]
      [filename yes|no]
      [object yes|no]
      [minmax yes|no]
      [frame yes|no]
      [image|physical|wcs|wcsa...wcsz yes|no]
      [red yes|no]
      [green yes|no]
      [blue yes|no]
```

Example:

```
$ds9 -view info no
```

```
$ds9 -view horzgraph yes
```

```
$ds9 -view wcsa yes
```

```
$ds9 -view red yes
$ds9 -view green no
$ds9 -view blue yes
```

visual

Force DS9 to use the specified color visual. This argument **MUST** be the first argument listed. Requires the visual be available.

Syntax:

```
-visual
```

```
[pseudocolor|pseudocolor8|truecolor|truecolor8|truecolor16|truecolor24]
```

Example:

```
$ds9 -visual truecolor24
```

VO

Invoke an connection to a Virtual Observatory site.

Syntax:

```
-vo <url>
```

Example:

```
$ds9 -vo chandra-ed.harvard.edu
```

WCS

Controls the World Coordinate System for the current frame. If the wcs system, skyframe, or skyformat is modified, the info panel, compass, grid, and alignment will be modified accordingly. Also, a new WCS specification can be loaded and used by the current image regardless of the WCS that was contained in the image file. Please see [WCS](#) for more information.

Syntax:

```
-wcs [wcs|wcsa...wcsz]
      [system wcs|wcsa...wcsz]
      [sky fk4|fk5|icrs|galactic|ecliptic]
      [skyformat degrees|sexagesimal]
      [align yes|no]
      [reset]
      [replace file <filename>]
      [append file <filename>]
```

Example:

```
$ds9 -wcs wcs
$ds9 -wcs system wcs
$ds9 -wcs sky fk5
$ds9 -wcs skyformat sexagesimal
$ds9 -wcs align yes
$ds9 -wcs reset
$ds9 -wcs replace file foo.wcs
$ds9 -wcs append file foo.wcs
```

web

Display specified URL in the web display.

`-web <url>`

Example:

```
$ds9 -web chandra.harvard.edu
```

xpa

Configure XPA at startup. The option `tcl` allows the user to send tcl code via xpa to be executed by ds9. Warning: this is a major security risk and is disabled by default. Please use with caution.

```
-xpa [yes|no]
      [inet|local|unix|localhost]
      [noxpans]
      [tcl yes|no]
```

Example:

```
$ds9 -xpa no
$ds9 -xpa local
$ds9 -xpa noxpans
$ds9 -xpa tcl yes
```

zmax

Set Scale Limits based on the *IRAF* algorithm and maximum data value.

Syntax:

```
-zmax
```

Example:

```
$ds9 -zmax
```

zscale

Set Scale Limits based on the *IRAF* algorithm.

Syntax:

```
-zscale
```

Example:

```
$ds9 -zscale
```

zoom

Controls the current zoom value for the current frame.

Syntax:

```
-zoom [<value>]
      [to <value>]
      [to fit]
```

Example:

```
$ds9 -zoom 2
$ds9 -zoom to 4
```

```
$ds9 -zoom to fit
```



XPA Access Points

The XPA messaging system provides seamless communication between DS9 and other Unix programs, including X programs, Perl, S-Lang, and Tcl/Tk. It also provides an easy way for users to communicate with DS9 by executing XPA client commands in the shell or by utilizing such commands in scripts. Because XPA works both at the programming level and the shell level, it is a powerful tool for unifying any analysis environment.

- about
- analysis
- array
- bin
- blink
- cd
- cmap
- contour
- crosshair
- cursor
- data
- datacube
- dss
- exit
- file
- fits
- frame
- grid
- iconify
- iis
- imexam
- lock
- lower
- match
- minmax
- mode
- nameserver
- orient
- page setup
- pan

- pixeltable
- plot
- prefs
- preserve
- print
- quit
- raise
- regions
- rgb
- rotate
- scale
- saveimage
- savefits
- savempeg
- shm
- single
- source
- tcl
- tile
- update
- version
- view
- vo
- wcs
- web
- zoom

about

Get DS9 credits.

Syntax:

about

Example:

```
$xpaget ds9 about
```

analysis

Control external analysis tasks. Tasks are numbered as they are loaded, starting with 1. Can also be used to display a message and display text in the text dialog window. To plot data, use the plot xpa point. Current tasks can be cleared via the clear command and new tasks can be loaded via the load command.

Syntax:

```
analysis [<task number>]
         [clear]
         [clear][load <filename>]
```

```
[message ok|okcancel|yesno {<message>}
[entry {<message>}]
[text]
```

Example:

```
$xpaget ds9 analysis
$xpaset -p ds9 analysis 0 # invoke first analysis task
$xpaset -p ds9 analysis clear
$xpaset -p ds9 analysis load my.analysis
$xpaset -p ds9 analysis clear load my.analysis
$xpaset -p ds9 analysis message ok '{This is a test}'
$xpaget ds9 analysis entry '{Please enter something}'
$cat my.analysis | xpaset ds9 analysis load
$cat foo.txt | xpaset ds9 analysis text
```

array

Load raw data array from stdin. If new is specified, a new frame is created first, before loading.

Syntax:

```
array
[new][[xdim=<x>,ydim=<y>|dim=<dim>],zdim=<z>,bitpix=<b>,skip=<s>,
      arch=[littleendian|bigendian]]
array [new] rgb
[[xdim=<x>,ydim=<y>|dim=<dim>],zdim=<z>,bitpix=<b>,skip=<s>,
      arch=[littleendian|bigendian]]
```

Example:

```
$xpaget ds9 array
$cat foo.arr | xpaset ds9 array [dim=512,bitpix=16]
$cat rgb.arr | xpaset ds9 array rgb [dim=200,zdim=3,bitpix=8]
$cat bar.arr | xpaset ds9 array [xdim=512,ydim=512,zdim=1,bitpix=16]
# load 512x512 short
$cat bar.arr | xpaset ds9 array [dim=256,bitpix=-32,skip=4] # load
256x256 float with 4 byte head
$cat bar.arr | xpaset ds9 array
[dim=512,bitpix=32,arch=littleendian] # load 512x512 long, intel
```

bin

Controls binning factor, binning buffer size, and binning function for binning FITS bin tables. The access point blocking is provided for backward compatibility.

Syntax:

```
bin [about <x> <y>]
    [buffer size <value>]
    [cols <x> <y>]
    [factor <value> [<vector>]]
    [depth <value>]
    [filter <string>]
```

```
[function average|sum]
[to fit]
[smooth yes|no]
[smooth function boxcar|tophat|gaussian]
[smooth radius <value>]
```

Example:

```
$xpaget ds9 bin about
$xpaget ds9 bin buffersize
$xpaget ds9 bin cols
$xpaget ds9 bin factor
$xpaget ds9 bin depth
$xpaget ds9 bin filter
$xpaget ds9 bin function
$xpaget ds9 bin smooth
$xpaget ds9 bin smooth function
$xpaget ds9 bin smooth radius
$xpaset -p ds9 bin about 4096 4096
$xpaset -p ds9 bin buffersize 512
$xpaset -p ds9 bin cols detx dety
$xpaset -p ds9 bin factor 4
$xpaset -p ds9 bin factor 4 2
$xpaset -p ds9 bin depth 10
$xpaset -p ds9 bin filter '{pha > 5}'
$xpaset -p ds9 bin function sum
$xpaset -p ds9 bin to fit
$xpaset -p ds9 bin smooth yes
$xpaset -p ds9 bin smooth boxcar
$xpaset -p ds9 bin smooth radius 3
```

blink

Select Blink Display Mode

Syntax:

```
blink
```

Example:

```
$xpaget ds9 blink
$xpaset -p ds9 blink
```

cd

Sets/Returns the current working directory.

Syntax:

```
cd [<directory>]
```

Example:

```
$xpaget ds9 cd
$xpaset -p ds9 cd /home/mrbill
```

cmap

Controls the colormap for the current frame. The colormap name is not case sensitive. A valid contrast value is from 0 to 10 and bias value from 0 to 1.

Syntax:

```
cmap [<colormap>]
      [file <filename>]
      [invert yes|no]
      [value <contrast> <bias>]
```

Example:

```
$xpaget ds9 cmap
$xpaget ds9 cmap file
$xpaget ds9 cmap invert
$xpaget ds9 cmap value
$xpaset -p ds9 cmap Heat
$xpaset -p ds9 cmap file foo.sao
$xpaset -p ds9 cmap invert yes
$xpaset -p ds9 cmap value 5 .5
```

contour

Controls contours in the current frame.

Syntax:

```
contour [yes|no]
        [<coordinate system> [<sky frame>]]
        [copy]
        [clear]
        [paste <coordinate system> [<sky frame>] <color> <width>]
        [load <filename> <coordinate system> [<sky frame>] <color>
<width>]
        [save <filename> <coordinate system> [<sky frame>]]
```

Example:

```
$xpaget ds9 contour
$xpaget ds9 contour wcs fk5
$xpaset -p ds9 contour yes
$xpaset -p ds9 contour copy
$xpaset -p ds9 contour clear
$xpaset -p ds9 contour paste wc fk4 red 2
$xpaset -p ds9 contour load foo.con wcs fk5 yellow 2
$xpaset -p ds9 contour save foo.con wcs fk5
```

crosshair

Controls the current position of the crosshair in the current frame. DS9 is placed in crosshair mode when the crosshair is set.

Syntax:

```
crosshair [x y <coordinate system> [<sky frame>][<sky format>]]
```

Example:

```
$xpaget ds9 crosshair # get crosshair in physical coords
$xpaget ds9 crosshair wcs fk4 sexagesimal # get crosshair in wcs
coords
$xpaset -p ds9 crosshair 100 100 physical # set crosshair in
physical
$xpaset -p ds9 crosshair 345 58.8 wcs fk5 # set crosshair in wcs
coords
$xpaset -p ds9 crosshair 23:01:00 +58:52:51 wcs fk5
```

cursor

Move mouse pointer or crosshair in image pixels in the current frame. Note, this will move selected Regions also.

Syntax:

```
cursor [x y]
```

Example:

```
$xpaset -p ds9 cursor 10 10
```

data

Return an array of data values given a lower left corner and a width and height in specified coordinate system. The last argument indicates if the coordinates are listed or just the data values.

Syntax:

```
data [<coordinate system> [<sky frame>] <x> <y> <w> <h> [yes|no]]
```

Example:

```
$xpaget ds9 data image 450 520 3 3 yes
$xpaget ds9 data physical 899 1039 6 6 no
$xpaget ds9 data fk5 202.47091 47.196811 0.00016516669 0.00016516669
no
$xpaget ds9 data wcs fk5 13:29:53.018 +47:11:48.52 0.00016516669
0.00016516669 no
```

datacube

Controls FITS datacube.

Syntax:

```
datacube [play|stop|next|prev|first|last]
        [#]
        [interval #]
```

Example:

```
$xpaget ds9 datacube
$xpaget ds9 datacube interval
$xpaset -p ds9 datacube play
$xpaset -p ds9 datacube last
$xpaset -p ds9 datacube 3
$xpaset -p ds9 datacube interval 2
```

dss

Support for Digital Sky Survey. The first three options will download an image based on the name or specified fk5 coordinate. The remaining options are used to configure downloads.

Syntax:

```
dss [<object>]
    [name <object>]
    [coordinate <ra> <dec>] # in wcs fk5

    [server sao|stsci|eso]
    [survey dss|dss2red|dss2blue]
    [size <width> <height>] # in arcmin
```

Example:

```
$xpaget ds9 dss name
$xpaget ds9 dss coord
$xpaget ds9 dss server
$xpaget ds9 dss survey
$xpaget ds9 dss size
$xpaset -p ds9 dss m31
$xpaset -p ds9 dss name m31
$xpaset -p ds9 dss coord 00:42:44.404 +41:16:08.78
$xpaset -p ds9 dss server eso
$xpaset -p ds9 dss survey dss2red
$xpaset -p ds9 dss size 10 10
```

exit

Exit DS9.

Syntax:

```
exit
```

Example:

```
$xpaset -p ds9 exit
```

file

file Load a FITS image, FITS Mosaic image(s), or array from a file into the current frame, or return the current file name(s) loaded for the current frame.

Syntax:

```
file [new][<filename>]
    [new][fits <filename>]
    [new][sfits <filename> <filename>]
    [new][medatacube <filename>]
    [new][mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] <filename>]
    [new][mosaicimagenext [wcs|wcsa...wcsz] <filename>]
    [new][mosaic [iraf|wcs|wcsa...wcsz] <filename>]
    [new][smosaic [iraf|wcs|wcsa...wcsz] <filename> <filename>]
```

```

[new][rgbcube <filename>]
[new][srgbcube <filename> <filename>]
[new][rgbimage <filename>]
[new][rgbarray
<filename>[[xdim=<x>,ydim=<y>|dim=<dim>],zdim=3,bitpix=<b>,[skip=<s>]]]
[new][array
<filename>[[xdim=<x>,ydim=<y>|dim=<dim>],zdim=<z>,bitpix=<b>,[skip=<s>]]]
[new][url <url>]
[save <filename>]
[save gz <filename>]
[save resample <filename>]
[save resample gz <filename>]

```

Example:

```

$xpaset ds9 file
$xpaset -p ds9 file foo.fits
$xpaset -p ds9 file fits foo.fits
$xpaset -p ds9 file sfits foo.hdr foo.arr
$xpaset -p ds9 file medatacube foo.fits
$xpaset -p ds9 file mosaicimage iraf bar.fits
$xpaset -p ds9 file mosaicimage wcs bar.fits
$xpaset -p ds9 file mosaicimage wcsa bar.fits
$xpaset -p ds9 file mosaicimage wfpc2 hst.fits
$xpaset -p ds9 file mosaicimagenext wcs bar.fits
$xpaset -p ds9 file mosaic iraf foo.fits
$xpaset -p ds9 file mosaic wcs bar.fits
$xpaset -p ds9 file smosaic iraf foo.hdr foo.arr
$xpaset -p ds9 file smosaic wcs bar.hdr bar.arr
$xpaset -p ds9 file rgbcube rgb.fits
$xpaset -p ds9 file srgbcube rgb.hdr rgb.arr
$xpaset -p ds9 file rgbimage rgb.fits
$xpaset -p ds9 file rgbarray rgb.arr[dim=200,zdim=3,bitpix=-32]
$xpaset -p ds9 file array array.arr[dim=512,bitpix=-32]
$xpaset -p ds9 file url 'ftp://foo.bar.edu/img.fits'
$xpaset -p ds9 file save foo.fits # save the current frame as FITS
Image
$xpaset -p ds9 file save gz foo.fits.gz # save as compressed FITS
Image
$xpaset -p ds9 file save resample foo.fits # save current
pan/zoom/rotate as FITS Image
$xpaset -p ds9 file save resample gz foo.fits.gz # save as
compressed FITS Image

```

fits

Load a FITS image from stdin into the current frame. Options can include the FITS extension or binning instructions. `xpaset` returns the FITS image in the current frame. If `new` is specified, a new frame is created before loading.

Syntax:

```
fits [size]
    [size [image|physical|wcs|wcsa...wcsz]
[degrees|arcmin|arcsecs]]
    [depth]
    [header [#] [keyword <string>]]
    [type]
    [image|table|resample] [gz]
    [new][<options>]
    [new][medatacube <options>]
    [new][mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] <options>]
    [new][mosaicimagenext [wcs|wcsa...wcsz] <options>]
    [new][mosaic [iraf|wcs|wcsa...wcsz] <options>]
    [new][rgbcube <options>]
    [new][rgbimage <options>]
    [save resample gz <filename>]
```

Example:

```
$xpaset ds9 fits > foo.fits
$xpaset ds9 fits size
$xpaset ds9 fits size wcs arcmin
$xpaset ds9 fits depth
$xpaset ds9 fits header # primary
$xpaset ds9 fits header 2 # hdu 2
$xpaset ds9 fits header -2 # hdu 2 with inherit
$xpaset ds9 fits header keyword "'BITPIX'"
$xpaset ds9 fits header 1 keyword "'BITPIX'"
$xpaset ds9 fits type
$xpaset ds9 fits image > foo.fits
$xpaset ds9 fits image gz > foo.fits.gz
$xpaset ds9 fits table > bar.fits
$xpaset ds9 fits table gz > bar.fits.gz
$xpaset ds9 fits resample > bar.fits
$xpaset ds9 fits resample gz > bar.fits.gz
$cat foo.fits | xpaset ds9 fits
$cat abc.fits | xpaset ds9 fits [2]
$cat bar.fits | xpaset ds9 fits new [bin=detx,dety]
$cat foo.fits | xpaset ds9 fits medatacube
$cat bar.fits | xpaset ds9 fits mosaicimage iraf
$cat bar.fits | xpaset ds9 fits mosaicimage wcs
$cat bar.fits | xpaset ds9 fits mosaicimage wcsa
$cat hst.fits | xpaset ds9 fits mosaicimage wfpc2
$cat bar.fits | xpaset ds9 fits mosaicimagenext wcs
```

```

$cat bar.fits | xpsaset ds9 fits mosaic iraf
$cat bar.fits | xpsaset ds9 fits mosaic wcs
$cat rgb.fits | xpsaset ds9 fits rgbcube
$cat rgb.fits | xpsaset ds9 fits rgbimage

```

frame

Controls frame functions. Frames may be created, deleted, reset, and centered. While return the current frame number. If you goto a frame that does not exists, it will be created. If the frame is hidden, it will be shown. The 'frameno' option is available for backward compatibility.

Syntax:

```

frame [center [#|all]]
      [clear [#|all]]
      [new [rgb]]
      [delete [#|all]]
      [reset [#|all]]
      [refresh [#|all]]
      [hide [#|all]]
      [show [#|all]]
      [first]
      [next]
      [prev]
      [last]
      [frameno #]
      [#]

```

Example:

```

$xpaset ds9 frame # returns the id of the current frame
$xpaset ds9 frame frameno # returns the id of the current frame
$xpaset ds9 frame all # returns the id of all frames
$xpaset ds9 frame active # returns the id of all active frames
$xpaset -p ds9 frame center # center current frame
$xpaset -p ds9 frame center 1 # center 'Frame1'
$xpaset -p ds9 frame center all # center all frames
$xpaset -p ds9 frame clear # clear current frame
$xpaset -p ds9 frame new # create new frame
$xpaset -p ds9 frame new rgb # create new rgb frame
$xpaset -p ds9 frame delete # delete current frame
$xpaset -p ds9 frame reset # reset current frame
$xpaset -p ds9 frame refresh # refresh current frame
$xpaset -p ds9 frame hide # hide current frame
$xpaset -p ds9 frame show 1 # show frame 'Frame1'
$xpaset -p ds9 frame first # goto first frame
$xpaset -p ds9 frame next # goto next frame
$xpaset -p ds9 frame prev # goto prev frame
$xpaset -p ds9 frame last # goto last frame
$xpaset -p ds9 frame frameno 4 # goto frame 'Frame4', create if
needed

```

```
$xpaset -p ds9 frame 3 # goto frame 'Frame3', create if needed
```

grid

Controls coordinate grid.

Syntax:

```
grid [yes|no]
      [load <filename>]
      [save <filename>]
      [system <coordinate system>]
      [sky <sky frame>]
      [skyformat <skyformat>]
      [type analysis|publication]
      [type axes interior|exterior]
      [type numerics interior|exterior]
      [view grid|axes|title|border|vertical yes|no]
      [view axes numbers|tickmarks|label yes|no]
```

Example:

```
$xpaget ds9 grid
$xpaget ds9 grid system
$xpaget ds9 grid sky
$xpaget ds9 grid skyformat
$xpaget ds9 grid type
$xpaget ds9 grid type axes
$xpaget ds9 grid type numerics
$xpaget ds9 grid view grid
$xpaget ds9 grid view axes
$xpaget ds9 grid view axes numbers
$xpaget ds9 grid view axes tickmarks
$xpaget ds9 grid view axes label
$xpaget ds9 grid view title
$xpaget ds9 grid view border
$xpaget ds9 grid view vertical
$xpaset -p ds9 grid yes
$xpaset -p ds9 grid load foo.grd
$xpaset -p ds9 grid save foo.grd
$xpaset -p ds9 grid system wcs
$xpaset -p ds9 grid sky fk5
$xpaset -p ds9 grid skyformat degrees
$xpaset -p ds9 grid type analysis
$xpaset -p ds9 grid type axes interior
$xpaset -p ds9 grid type numerics interior
$xpaset -p ds9 grid view grid yes
$xpaset -p ds9 grid view axes yes
$xpaset -p ds9 grid view axes numbers yes
$xpaset -p ds9 grid view axes tickmarks yes
```

```
$xpaget -p ds9 grid view axes label yes
$xpaget -p ds9 grid view title yes
$xpaget -p ds9 grid view border yes
$xpaget -p ds9 grid view vertical no
```

iconify

Toggles iconification.

Syntax:

```
iconify [yes|no]
```

Example:

```
$xpaget ds9 iconify
$xpaget -p ds9 iconify yes
```

iis

Set/Get IIS Filename.

Syntax:

```
iis [filename <filename> [#]]
```

Example:

```
$xpaget ds9 iis filename
$xpaget ds9 iis filename 4
$xpaget -p ds9 iis filename foo.fits
$xpaget -p ds9 iis filename bar.fits 4
```

imexam

Interactive examine function. A blinking cursor will indicate to the user to click on a point on an image. The specified information will be returned at that time.

Syntax:

```
imexam [coordinate <coordinate system> [<sky frame>] [<sky format>]]
      [data [width][height]]
```

Example:

```
$xpaget ds9 imexam coordinate image
$xpaget ds9 imexam coordinate wcs fk5 degrees
$xpaget ds9 imexam coordinate wcs galactic sexagesimal
$xpaget ds9 imexam coordinate fk5
$xpaget ds9 imexam data # return data value
$xpaget ds9 imexam data 3 3 # return all data in 3x3 box about
selected point
```

lock

Lock frames.

Syntax:

```
lock [crosshairs none|wcs|wcsa...wcsz|physical|image]
```

Example:

```
$xpaset -p ds9 lock crosshairs wcs
```

lower

Will lower ds9 in the window stacking order.

Syntax:

```
lower
```

Example:

```
$xpaset -p ds9 lower
```

match

Match all other frames to the current frame.

Syntax:

```
match [frames wcs|physical|image]
      [colorbars]
      [scales]
```

Example:

```
$xpaset -p ds9 match frames wcs
```

```
$xpaset -p ds9 match colorbars
```

```
$xpaset -p ds9 match scales
```

minmax

This is how DS9 determines the min and max data values from the data. SCAN will scan all data. SAMPLE will sample the data every n samples. DATAMIN and IRAFMIN will use the values of the keywords if present. In general, it is recommended to use SCAN unless your computer is slow or your data files are very large. Select the increment interval for determining the min and max data values during sampling. The larger the interval, the quicker the process.

Syntax:

```
minmax [scan|sample|datamin|irafmin]
       [mode scan|sample|datamin|irafmin]
       [interval <value>]
```

Example:

```
$xpaset ds9 minmax mode
```

```
$xpaset ds9 minmax interval
```

```
$xpaset -p ds9 minmax scan
```

```
$xpaset -p ds9 minmax mode sample
```

```
$xpaset -p ds9 minmax interval 10
```

mode

Controls the first mouse button mode.

Syntax:

```
mode [none|pointer|crosshair|colorbar|pan|zoom|rotate|examine]
```

Example:

```
$xpaset ds9 mode
```

```
$xpaset -p ds9 mode crosshair
```

nameserver

Support Name Server functions. Coordinates are in fk5.

Syntax:

```
nameserver [<object>]
           [name <object>]
           [server ned-sao|ned-eso|simbad-sao|simbad-eso]
           [skyformat degrees|sexagesimal]
```

Example:

```
$xpaget ds9 nameserver server
$xpaget ds9 nameserver skyformat
$xpaset -p ds9 nameserver m31
$xpaset -p ds9 nameserver name m31
$xpaset -p ds9 nameserver server ned-sao
$xpaset -p ds9 nameserver skyformat sexagesimal
```

orient

Controls the orientation of the current frame.

Syntax:

```
orient [none|x|y|xy]
```

Example:

```
$xpaget ds9 orient
$xpaset -p ds9 orient xy
```

page setup

Controls Page Setup options.

Syntax:

```
page setup [orientation portrait|landscape]
           [pagescale scaled|fixed]
           [pagesize letter|legal|tabloid|poster|a4]
```

Example:

```
$xpaget ds9 page setup orientation
$xpaget ds9 page setup pagescale
$xpaget ds9 page setup pagesize
$xpaset -p ds9 page setup orientation portrait
$xpaset -p ds9 page setup pagescale scaled
$xpaset -p ds9 page setup pagesize poster
```

pan

Controls the current image cursor location for the current frame.

Syntax:

```
pan [x y <coordinate system> [<sky frame>][<sky format>]]
    [to x y <coordinate system> [sky frame]][<sky format>]
```

Example:

```
$xpaget ds9 pan # get current image coords
$xpaget ds9 pan wcs fk4 sexagesimal # get current wcs coords
$xpaset -p ds9 pan 200 200 image # pan relative
$xpaset -p ds9 pan to 400 400 physical # pan to physical coords
$xpaset -p ds9 pan to 13:29:55 47:11:50 wcs fk5 # pan to wcs coords
```

pixeltable

Display/Hide the pixel table.

Syntax:

```
pixeltable [yes|no]
```

Example:

```
$xpaget ds9 pixeltable
$xpaset -p ds9 pixeltable yes
```

plot

Display and configure data plots. All plot commands take an optional second command, the plot name. Use `xpaget plot` to retrieve all plot names. If no plot name is specified, the last plot created is assumed. Plot data is assumed to be a pair of coordinates, with optional error values. The following are valid data descriptions:

<code>xy</code>	x and y coordinates
<code>xyex</code>	x,y coordinates with x errors
<code>xyey</code>	x,y coordinates with y errors
<code>xyexey</code>	x,y coordinates with both x and y errors

To create a new plot, use the `plot new` command. If the second arg is `stdin`, the title, x axis title, y axis title, and dimension are assumed to be on the first line of the data.

Syntax:

```
# create new empty plot window
plot []
    [new [name <plotname>]]
# create new plot with data
plot [new [name <plotname>] stdin]
    [new [name <plotname>] <title> <xaxis label> <yaxis label>
    xy|xyex|xyey|xyexey]
# load additional dataset into an existing plot
plot [<plotname>] [data xy|xyex|xyey|xyexey]
# edit existing plot
```

```

plot [<plotname>] [close]
    [<plotname>] [clear]
    [<plotname>] [load <filename> xy|xyex|xyey|xyexey]
    [<plotname>] [save <filename>]
    [<plotname>] [loadconfig <filename>]
    [<plotname>] [saveconfig <filename>]
    [<plotname>] [print]
    [<plotname>] [printdestination printer|file]
    [<plotname>] [print command <command>]
    [<plotname>] [print filename <filename>]
    [<plotname>] [print palette color|gray|mono]
    [<plotname>] [page orientation portrait|landscape]
    [<plotname>] [page pagescale scaled|fixed]
    [<plotname>] [page pagesize letter|legal|tabloid|poster|a4]
    [<plotname>] [graph grid yes|no]
    [<plotname>] [graph scale]
linearlinear|linearlog|loglinear|loglog]
    [<plotname>] [graph range x|y auto yes|no]
    [<plotname>] [graph range x|y min <value>]
    [<plotname>] [graph range x|y max <value>]
    [<plotname>] [graph labels title|xaxis|yaxis <value>]
    [<plotname>] [font numbers|labels|title font]
times|helvetica|symbol|courier]
    [<plotname>] [font numbers|labels|title size <value>]
    [<plotname>] [font numbers|labels|title style]
plain|bold|italic]
# edit current dataset
plot [<plotname>] [dataset #]
    [<plotname>] [view discrete|line|step|quadratic|errorbar
yes|no]
    [<plotname>] [color]
discrete|line|step|quadratic|errorbar<color>]
    [<plotname>] [line discrete circle|diamond|plus|cross]
    [<plotname>] [line line|step|quadratic|errorbar width <value>]
    [<plotname>] [line line|step|quadratic dash yes|no]
    [<plotname>] [line errorbar style 1|2]

```

Example:

```

$xpaget ds9 plot # return all plotnames
# create new empty plot window
$xpaset -p ds9 plot
$xpaset -p ds9 plot new
$xpaset -p ds9 plot new name foo
# create new plot with data
$cat foo.dat | xpaset ds9 plot new stdin
$cat foo.dat | xpaset ds9 plot new name foo stdin
$cat bar.dat | xpaset ds9 plot new "{The Title}" "{X}" "{Y}" xy
$cat bar.dat | xpaset ds9 plot new name foo "{The Title}" "{X}"

```

```

"{Y}" xy
# load additional dataset into an existing plot
$cat bar.dat | xpaset ds9 plot data xy # plot additional data
$cat bar.dat | xpaset ds9 plot foo data xy # plot additional data
# edit existing plot
$xpaset -p ds9 plot close # close last plot
$xpaset -p ds9 plot foo close # close plot foo
$xpaset -p ds9 plot clear # clear all datasets
$xpaset -p ds9 plot load foo.dat xy # load new dataset with
dimension xy
$xpaset -p ds9 plot save bar.dat # save current dataset
$xpaset -p ds9 plot loadconfig foo.plt # load plot configuration
$xpaset -p ds9 plot saveconfig bar.plt # save current plot
configuration
$xpaset -p ds9 plot print
$xpaset -p ds9 plot print destination file
$xpaset -p ds9 plot print command "lp"
$xpaset -p ds9 plot print filename "foo.ps"
$xpaset -p ds9 plot print palette gray
$xpaset -p ds9 plot page orientation portrait
$xpaset -p ds9 plot page pagescale scaled
$xpaset -p ds9 plot page pagesize letter
$xpaset -p ds9 plot graph grid yes
$xpaset -p ds9 plot graph scale loglog
$xpaset -p ds9 plot graph range x auto yes
$xpaset -p ds9 plot graph range x min 0
$xpaset -p ds9 plot graph range x max 100
$xpaset -p ds9 plot graph range y auto yes
$xpaset -p ds9 plot graph range y min 0
$xpaset -p ds9 plot graph range y max 100
$xpaset -p ds9 plot graph labels title {The Title}
$xpaset -p ds9 plot graph labels xaxis {X}
$xpaset -p ds9 plot graph labels yaxis {Y}
$xpaset -p ds9 plot font numbers font times
$xpaset -p ds9 plot font numbers size 12
$xpaset -p ds9 plot font numbers style bold
$xpaset -p ds9 plot font labels font times
$xpaset -p ds9 plot font title font times
# edit current dataset
$xpaset -p ds9 plot dataset 2 # set current dataset to the second
dataset loaded
$xpaset -p ds9 plot view discrete yes
$xpaset -p ds9 plot color discrete red
$xpaset -p ds9 plot line discrete cross
$xpaset -p ds9 plot line step width 2
$xpaset -p ds9 plot line step dash yes
$xpaset -p ds9 plot line errorbar style 2

```

prefs

Controls various preference settings.

Syntax:

```
prefs [mosaicfast yes|no]
      [bgcolor white|black|red|green|blue|cyan|magenta|yellow]
      [nancolor white|black|red|green|blue|cyan|magenta|yellow]
```

Example:

```
$xpaget ds9 prefs mosaicfast
$xpaget ds9 prefs bgcolor
$xpaget ds9 prefs nancolor
$xpaget ds9 prefs wcsprojection
$xpaset -p ds9 prefs mosaicfast no
$xpaset -p ds9 prefs bgcolor black
$xpaset -p ds9 prefs nancolor red
```

preserve

Preserve the follow attributes while loading a new image.

Syntax:

```
preserve [scale yes|no]
         [pan yes|no]
         [regions yes|no]
```

Example:

```
$xpaget ds9 preserve scale
$xpaget ds9 preserve pan
$xpaget ds9 preserve regions
$xpaset -p ds9 preserve scale yes
$xpaset -p ds9 preserve pan yes
$xpaset -p ds9 preserve regions yes
```

print

Controls printing. Use `print` option to set printing options. Use `print` to actually print.

Syntax:

```
print [destination printer|file]
      [command <command>]
      [filename <filename>]
      [palette rgb|cmyk|gray]
      [level 1|2]
      [interpolate yes|no]
      [resolution 53|72|75|150|300|600]
```

Example:

```
$xpaget ds9 print destination
$xpaget ds9 print command
$xpaget ds9 print filename
$xpaget ds9 print palette
```

```

$xpaget ds9 print level
$xpaget ds9 print interpolate
$xpaget ds9 print resolution
$xpaset -p ds9 print
$xpaset -p ds9 print destination file
$xpaset -p ds9 print command '{gv -}'
$xpaset -p ds9 print filename foo.ps
$xpaset -p ds9 print palette cmyk
$xpaset -p ds9 print level 2
$xpaset -p ds9 print interpolate no
$xpaset -p ds9 print resolution 75

```

quit

Exit DS9.

Syntax:

```
quit
```

Example:

```
$xpaset -p ds9 quit
```

raise

Will raise ds9 in the window stacking order.

Syntax:

```
raise
```

Example:

```
$xpaset -p ds9 raise
```

regions

Controls regions in the current frame.

Syntax:

```

regions [fg|bg] [move front]
        [fg|bg] [move back]
        [fg|bg] [select all]
        [fg|bg] [select none]
        [fg|bg] [select group <groupname>]
        [fg|bg] [delete all]
        [fg|bg] [delete select]
        [fg|bg] [load <filename>]
        [fg|bg] [save <filename>]
        [format]
        [-format ds9|ciao|saotng|saoimage|pros|xy]
        [system]
        [-system image|physical|wcs|wcsa...wcsz]
        [sky]
        [-sky fk4|fk5|icrs|galactic|ecliptic]

```

```

    [skyformat]
    [-skyformat degrees|sexagesimal]
    [strip]
    [-strip yes|no]
[wcs]
[-wcs]
    [shape <shape>]
    [color white|black|red|green|blue|cyan|magenta|yellow]
    [width <width>]
    [delim [nl|<char>]]
    [fg|bg] [include|exclude|source|background|selected]
    [fg|bg] [-prop]
select|edit|move|rotate|delete|fixed|include|source 1|0]
    [fg|bg] [groups]
    [fg|bg] [-group <tag>]
    [fg|bg] [group <tag> color
[black|white|red|green|blue|cyan|magenta|yellow]]
    [fg|bg] [group <tag> copy]
    [fg|bg] [group <tag> delete]
    [fg|bg] [group <tag> cut]
    [fg|bg] [group <tag> font <font>]
    [fg|bg] [group <tag> move <int> <int>]
    [fg|bg] [group <tag> movefront]
    [fg|bg] [group <tag> moveback]
    [fg|bg] [group <tag> property <property> [yes|no]]
    [fg|bg] [group <tag> select]

```

Example:

```

$xpaget ds9 regions
$xpaget ds9 regions bg
$xpaget ds9 regions -format ds9 -coord wcs -sky fk5 -skyformat
sexagesimal -prop edit 1 -group foo
$xpaget ds9 regions format
$xpaget ds9 regions system
$xpaget ds9 regions sky
$xpaget ds9 regions skyformat
$xpaget ds9 regions strip
$xpaget ds9 regions wcs
$xpaget ds9 regions shape
$xpaget ds9 regions color
$xpaget ds9 regions width
$xpaget ds9 regions delim
$xpaget ds9 regions source
$xpaget ds9 regions background
$xpaget ds9 regions include
$xpaget ds9 regions exclude
$xpaget ds9 regions selected
$xpaget ds9 regions -prop select yes

```

```

$xpaset ds9 regions groups
$xpaset ds9 regions -group foo
$echo "image; circle 100 100 20" | xpaset ds9 regions
$echo "image; circle 100 100 20" | xpaset ds9 regions bg
$echo "fk5; circle 13:29:55 47:11:50 .5'" | xpaset ds9 regions
$echo "physical; ellipse 100 100 20 40" | xpaset ds9 regions
$echo "box 100 100 20 40 25" | xpaset ds9 regions
$echo "image; line 100 100 200 400" | xpaset ds9 regions
$echo "physical; ruler 200 300 200 400" | xpaset ds9 regions
$echo "image; text 100 100 # text={Hello, World}" | xpaset ds9
regions
$echo "fk4; boxcircle point 13:29:55 47:11:50" | xpaset ds9 regions
$xpaset -p ds9 regions move back
$xpaset -p ds9 regions move front
$xpaset -p ds9 regions select all
$xpaset -p ds9 regions select none
$xpaset -p ds9 regions select group foo
$xpaset -p ds9 regions delete all
$xpaset -p ds9 regions delete select
$xpaset -p ds9 regions load foo.reg
$xpaset -p ds9 regions load regions.fits
$xpaset -p ds9 regions save foo.reg
$xpaset -p ds9 regions format ds9
$xpaset -p ds9 regions system wcs
$xpaset -p ds9 regions sky fk5
$xpaset -p ds9 regions skyformat degrees
$xpaset -p ds9 regions delim nl
$xpaset -p ds9 regions strip yes
$xpaset -p ds9 regions wcs yes
$xpaset -p ds9 regions shape ellipse
$xpaset -p ds9 regions color red
$xpaset -p ds9 regions width 3
$xpaset -p ds9 regions group foo color red
$xpaset -p ds9 regions group foo copy
$xpaset -p ds9 regions group foo delete
$xpaset -p ds9 regions group foo cut
$xpaset -p ds9 regions group foo font {times 14 bold}
$xpaset -p ds9 regions group foo move 100 100
$xpaset -p ds9 regions group foo movefront
$xpaset -p ds9 regions group foo moveback
$xpaset -p ds9 regions group foo property delete no
$xpaset -p ds9 regions group foo select

```

rgb

Create RGB frame and control RGB frame parameters.

Syntax:

```
rgb # empty
    [red|green|blue]
    [channel [red|green|blue]]
    [view [red|green|blue] [yes|no]]
    [system <coordinate system>]
    [lock scale|bin|colorbar [yes|no]]
```

Example:

```
$xpaget ds9 rgb channel
$xpaget ds9 rgb lock bin
$xpaget ds9 rgb lock scale
$xpaget ds9 rgb lock colorbar
$xpaget ds9 rgb system
$xpaget ds9 rgb view red
$xpaget ds9 rgb view green
$xpaget ds9 rgb view blue
$xpaset -p ds9 rgb # create new rgb frame
$xpaset -p ds9 rgb red # set current channel to red
$xpaset -p ds9 rgb channel red # set current channel to red
$xpaset -p ds9 rgb view blue no # turn off blue channel
$xpaset -p ds9 rgb system wcs # set rgb coordinate system
$xpaset -p ds9 rgb lock scale yes # lock rgb channels for scaling
$xpaset -p ds9 rgb lock bin yes # lock rgb channels for binning
$xpaset -p ds9 rgb lock colorbar yes # lock rgb colorbar channels
```

rotate

Controls the rotation angle (in degrees) of the current frame.

Syntax:

```
rotate [<value>]
      [to <value>]
```

Example:

```
$xpaget ds9 rotate
$xpaset -p ds9 rotate 45
$xpaset -p ds9 rotate to 30
```

saveimage

Save visible image(s) as a raster. If image is a data cube, the mpeg option will cycle thru each slice creating a mpeg movie.

Syntax:

```
saveimage [fits|jpeg|tiff|png|ppm|mpeg <filename>]
```

Example:

```
$xpaset -p ds9 saveimage jpeg ds9.jpg
```

savefits

Save current frame data as FITS. This differs from SAVEIMAGE in that the entire image of the current frame is saved as a FITS, without graphics.

Syntax:

```
savefits [<filename>]
```

Example:

```
$xpaset -p ds9 savefits ds9.fits
```

savempeg

Save all active frames as a mpeg movie.

Syntax:

```
savempeg [<filename>]
```

Example:

```
$xpaset -p ds9 savempeg ds9.mpg
```

scale

Controls the limits, color scale distribution, and use of DATASEC keyword.

Syntax:

```
scale [linear|log|squared|sqrt|histequ]
      [datasec yes|no]
      [limits <minvalue> <maxvalue>]
      [mode minmax|<value>|zscale|zmax]
      [scope local|global]
```

Example:

```
$xpaset ds9 scale
$xpaset ds9 scale datasec
$xpaset ds9 scale limits
$xpaset ds9 scale mode
$xpaset ds9 scale scope
$xpaset -p ds9 scale datasec yes
$xpaset -p ds9 scale histequ
$xpaset -p ds9 scale limits 1 100
$xpaset -p ds9 scale mode zscale
$xpaset -p ds9 scale mode 99.5
$xpaset -p ds9 scale scope local
```

shm

Load a shared memory segment into the current frame.

Syntax:

```
shm [<key> [<filename>]]
    [key <key> [<filename>]]
    [shmid <id> [<filename>]]
    [fits [key|shmid] <id> [<filename>]]
```

```

    [sfits [key|shmid] <id> <id> [<filename>]]
    [mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] [key|shmid] <id>
[<filename>]]
    [mosaicimagenext [wcs|wcsa...wcsz] [key|shmid] <id>
[<filename>]]
    [mosaic [iraf|wcs|wcsa...wcsz] [key|shmid] <id> [<filename>]]
    [smosaic [iraf|wcs|wcsa...wcsz] [key|shmid] <id> [<filename>]]
    [rgbcube [key|shmid] <id> [<filename>]]
    [srgbcube [key|shmid] <id> [<filename>]]
    [rgbimage [key|shmid] <id> [<filename>]]
    [rgbarray [key|shmid] <id>
[xdim=<x>,ydim=<y>|dim=<dim>,zdim=3],bitpix=<b>,[skip=<s>]]
    [array [key|shmid] <id>
[xdim=<x>,ydim=<y>|dim=<dim>],bitpix=<b>,[skip=<s>]]
    [startload|finishload]

```

Example:

```

$xpaset ds9 shm
$xpaset -p ds9 shm 102
$xpaset -p ds9 shm key 102
$xpaset -p ds9 shm shmid 102 foo
$xpaset -p ds9 shm fits key 100 foo
$xpaset -p ds9 shm sfits key 100 101 foo
$xpaset -p ds9 shm mosaicimage iraf key 100 foo
$xpaset -p ds9 shm mosaicimage wcs key 100 foo
$xpaset -p ds9 shm mosaicimage wcsa key 100 foo
$xpaset -p ds9 shm mosaicimage wfpc2 key 100 foo
$xpaset -p ds9 shm mosaicimagenext wcs key 100 foo
$xpaset -p ds9 shm mosaic iraf key 100 foo
$xpaset -p ds9 shm mosaic wcs key 100 foo
$xpaset -p ds9 shm smosaic wcs key 100 101 foo
$xpaset -p ds9 shm rgbcube key 100 foo
$xpaset -p ds9 shm srgbcube key 100 101 foo
$xpaset -p ds9 shm rgbimage key 100 foo
$xpaset -p ds9 shm rgbarray key 100 [dim=200,zdim=3,bitpix=-32]
$xpaset -p ds9 shm array shmid 102 [dim=32,bitpix=-32]
$xpaset -p ds9 shm startload # start a multiple load sequence
without updating the display
$xpaset -p ds9 shm finishload # finish multiple load sequence

```

single

Select Single Display mode

Syntax:

```
single
```

Example:

```

$xpaset ds9 single
$xpaset -p ds9 single

```

source

Source tcl code from a file.

Syntax:

```
source [filename]
```

Example:

```
$xpaset -p ds9 source foo.tcl
```

tcl

Execute one tcl command

Syntax:

```
tcl [<tcl command>]
```

Example:

```
$echo 'puts "Hello, World"' | xpaset ds9 tcl
```

tile

Controls the tile display mode.

Syntax:

```
tile [yes|no]
      [mode grid|column|row]
      [grid]
      [grid mode [automatic|manual]]
      [grid layout <row> <col>]
      [grid gap <pixels>]
      [row]
      [column]
```

Example:

```
$xpaset ds9 tile
$xpaset ds9 tile mode
$xpaset ds9 tile grid mode
$xpaset ds9 tile grid layout
$xpaset ds9 tile grid gap
$xpaset -p ds9 tile
$xpaset -p ds9 tile mode row
$xpaset -p ds9 tile grid
$xpaset -p ds9 tile grid mode manual
$xpaset -p ds9 tile grid layout 5 5
$xpaset -p ds9 tile grid gap 10
$xpaset -p ds9 tile row
$xpaset -p ds9 tile column
```

update

Updates the current frame or region of frame. In the second form, the first argument is the number of the fits HDU (starting with 1) and the remaining args are a bounding box in IMAGE coordinates. By default, the screen is updated the next available idle cycle. However, you may force an immediate update by specifying the NOW option.

Syntax:

```
update [ ]
        [# x1 y1 x2 y2]
        [now]
        [now # x1 y1 x2 y2]
        [on]
        [off]
```

Example:

```
$xpaset -p ds9 update
$xpaset -p ds9 update 1 100 100 300 400
$xpaset -p ds9 update now
$xpaset -p ds9 update now 1 100 100 300 400
$xpaset -p ds9 update off # delay refresh of the screen while
loading files
$xpaset -p ds9 update on # be sure to turn it on when you are
finished loading
```

version

Returns the current version of DS9.

Syntax:

```
version
```

Example:

```
$xpaset ds9 version
```

view

Controls the GUI.

Syntax:

```
view [info yes|no]
     [panner yes|no]
     [magnifier yes|no]
     [buttons yes|no]
     [colorbar yes|no]
     [horzgraph yes|no]
     [vertgraph yes|no]
     [filename yes|no]
     [object yes|no]
     [minmax yes|no]
     [frame yes|no]
     [image|physical|wcs|wcsa...wcsz yes|no]
```

```
[red yes|no]
[green yes|no]
[blue yes|no]
```

Example:

```
$xpaget ds9 view info
$xpaget ds9 view horzgraph
$xpaget ds9 view wcsa
$xpaset -p ds9 view info no
$xpaset -p ds9 view horzgraph yes
$xpaset -p ds9 view wcsa yes
$xpaset -p ds9 view red yes
$xpaset -p ds9 view green no
$xpaset -p ds9 view blue yes
```

VO

Invoke an connection to a Virtual Observatory site.

Syntax:

```
vo <url>
```

Example:

```
$xpaget ds9 vo
$xpaset -p ds9 vo chandra-ed.harvard.edu
```

WCS

Controls the World Coordinate System for the current frame. If the wcs system, skyframe, or skyformat is modified, the info panel, compass, grid, and alignment will be modified accordingly. Also, using this access point, a new WCS specification can be loaded and used by the current image regardless of the WCS that was contained in the image file. WCS specification can be sent to DS9 as an ASCII file. Please see [WCS](#) for more information.

Syntax:

```
wcs [wcs|wcsa...wcsz]
    [system wcs|wcsa...wcsz]
    [sky fk4|fk5|icrs|galactic|ecliptic]
    [skyformat degrees|sexagesimal]
    [align yes|no]
    [reset]
    [replace file <filename>]
    [append file <filename>]
    [replace]
    [append]
```

Example:

```
$xpaget ds9 wcs
$xpaget ds9 wcs system
$xpaget ds9 wcs sky
$xpaget ds9 wcs skyformat
```

```
$xpaget ds9 wcs align
$xpaset -p ds9 wcs wcs
$xpaset -p ds9 wcs system wcs
$xpaset -p ds9 wcs sky fk5
$xpaset -p ds9 wcs skyformat sexagesimal
$xpaset -p ds9 wcs align yes
$xpaset -p ds9 wcs reset
$xpaset -p ds9 wcs replace file foo.wcs
$xpaset -p ds9 wcs append file foo.wcs
$cat foo.wcs | xpaset ds9 wcs replace
$cat foo.wcs | xpaset ds9 wcs append
$echo "OBJECT = 'foobar'" | xpaset ds9 wcs append
```

web

Display specified URL in the web display.

Syntax:

```
web <url>
```

Example:

```
$xpaget ds9 web
```

```
$xpaset -p ds9 web <url>
```

zoom

Controls the current zoom value for the current frame.

Syntax:

```
zoom [<value>]
      [to <value>]
      [to fit]
```

Example:

```
$xpaget ds9 zoom
```

```
$xpaset -p ds9 zoom 2
```

```
$xpaset -p ds9 zoom to 4
```

```
$xpaset -p ds9 zoom to fit
```



Regions

Regions provide a means for marking particular areas of an image for further analysis. Regions may also be used for presentation purposes. DS9 supports a number of region descriptions, each of which may be edited, moved, rotated, displayed, saved and loaded, via the GUI and XPA.

[Region Descriptions](#)

[Region Properties](#)

[Region File Format](#)

[External Region Files](#)

Region Descriptions

Circle

Usage: circle x y radius

Ellipse

Usage: ellipse x y radius radius angle

Box

Usage: box x y width height angle

Polygon

Usage: polygon x1 y1 x2 y2 x3 y3 ...

Line

Usage: line x1 y1 x2 y2 # line=[0|1] [0|1]

Text

Usage: text x y # text={Your Text Here}
text x y {Your Text Here}

Ruler

Usage: ruler x1 y1 x2 y2 # ruler=[pixels|degrees|arcmin|arcsec]

Compass

Usage: compass x1 y1 length # compass=<coordinate system> <north label> <east label> [0|1] [0|1]

Projection

Usage: projection x1 y1 x2 y2 width

Annulus

Usage: annulus x y inner outer n=#
annulus x y r1 r2 r3...

Ellipse Annulus

Usage: ellipse x y r11 r12 r21 r22 n=# [angle]
ellipse x y r11 r12 r21 r22 r31 r32 ... [angle]

Box Annulus

```
Usage: box x y w1 h1 w2 h2 [angle]
       box x y w1 h1 w2 h2 w3 h3 ... [angle]
```

Panda

```
Usage: panda x y startangle stopangle nangle inner outer nradius
```

Circle Point

```
Usage: point x y # point=circle
       circle point x y
```

Box Point

```
Usage: point x y # point=box
       box point x y
```

Diamond Point

```
Usage: point x y # point=diamond
       diamond point x y
```

Cross Point

```
Usage: point x y # point=cross
       cross point x y
```

X Point

```
Usage: point x y # point=x
       x point x y
```

Arrow Point

```
Usage: point x y # point=arrow
       arrow point x y
```

BoxCircle Point

```
Usage: point x y # point=boxcircle
       boxcircle point x y
```

Region Properties

Each region has a number of properties associated with the region, which indicates how the region is to be rendered or manipulated. Properties are defined for a region in the comment section of the region description. The exception is the Include/Exclude property. It is set via '+' or '-' preceding the region. In addition, the Line, Point, and Ruler regions have unique properties, not shared by others. Not all properties are available via the GUI or are applicable for all regions.

Text

All regions may have text associated with them. Use the text property to set the text. Strings may be quoted with " or ' or {}. For best results, use {}.

```
Example: circle(100,100,20) # text = {This message has both a " and
' in it}
```

Color

The color property specifies the color of the region when rendered. The follow 8 colors are supported:

- white
- black
- red
- green
- blue
- cyan
- magenta
- yellow

Example: `circle(100,100,20) # color = green`

Font

The font property specifies the font family, size, and type of any text to be displayed along with the region.

Example: `circle(100,100,20) # font="times 12 bold"`

Can Select

The Select property specifies if the user is allowed to select (hence, edit) the region via the GUI. For Regions used for catalogs and such, it is desirable that the user is unable to edit, move, or delete the region.

Example: `circle(100,100,20) # select = 1`

Can Edit

The Edit property specifies if the user is allowed to edit the region via the GUI.

Example: `circle(100,100,20) # edit = 1`

Can Move

The Move property specifies if the user is allowed to move the region via the GUI.

Example: `circle(100,100,20) # move = 1`

Can Rotate

The Rotate property specifies if the user is allowed to rotate the region via the GUI.

Example: `circle(100,100,20) # rotate = 1`

Can Delete

The Delete property specifies if the user is allowed to delete the region via the GUI.

Example: `circle(100,100,20) # delete = 1`

Can Highlight

The Highlight property specifies if the edit handles become visible when the region is selected.

Include/Exclude

The Include/Exclude properties flag the region with a boolean NOT for later analysis. Use '+' for include (default), '-' for exclude.

Example: `-circle(100,100,20)`

Source/Background

The Source/Background properties flag the region for use with other analysis applications. The default is source

Example: `circle(100,100,20) # source`
`circle(200,200,10) # background`

Fixed in Size

The Fixed in Size property specifies that the region does not change in size as the image magnification factor changes. This allows the user to build complex pointer type regions.

Example: `circle(100,100,20) # fixed = 1`

Tag

All regions may have zero or more tags associated with it, which may be used for grouping and searching.

Example: `circle(100,100,20) # tag = {Group 1} tag = {Group 2}`

Line

The line region may be rendered with arrows, one at each end. To indicate arrows, use the line property. A '1' indicates an arrow, '0' indicates no arrow.

Example: `line(100,100,200,200) # line= 1 1`

Ruler

The ruler region may display information in 'pixels', 'degrees', 'arcmin', or 'arcsec'. Use the ruler property to indicate which format to display distances in.

Example: `ruler(100,100,200,200) # ruler=arcmin`

Point

Point regions have an associated type. Use the point property to set the point type.

Example: `point(100,100) # point=diamond`

Default Properties

The default properties are:

```
text={}  
color=green  
font="helvetica 10 normal"  
select=1  
edit=1  
move=1  
delete=1  
highlite=1  
include=1  
fixed=0
```

Region File Format

Syntax

Region arguments may be separated with either a comma or space. Optional parentheses may be used at the beginning and end of a description.

```
circle 100 100 10  
circle(100 100 10)  
circle(100,100,10)
```

Comments

All lines that begin with # are comments and will be ignored.

```
# This is a comment
```

Delimiter

All lines may be delimited with either a new-line or semi-colon.

```
circle 100 100 10  
ellipse 200 200 20 40 ; box 300 300 20 40
```

Header

A DS9 region file may start with the following optional header:

```
# Region file format: DS9 version 3.0
```

Global Properties

Global properties affect all regions unless a local property is specified. The `global` keyword is first, followed by a list of `keyword = value` pairs. Multiple global property lines may be used within a region file.

```
global color=green font="helvetica 10 normal" edit=1 move=1
delete=1 hlite=1 include=1 wcs=wcs
```

Local Properties

Local properties start with a `#` after a region description and only affect the region it is specified with.

```
physical;circle(504,513,20) # color=red text={This is a Circle}
```

Coordinate Systems

For each region, it is important to specify the coordinate system used to interpret the region, i.e., to set the context in which the position and size values are interpreted. For this purpose, the following keywords are recognized:

```
PHYSICAL          # pixel coords of original file using
LTM/LTV
IMAGE             # pixel coords of current file
FK4, B1950        # sky coordinate systems
FK5, J2000        # sky coordinate systems
GALACTIC          # sky coordinate systems
ECLIPTIC          # sky coordinate systems
ICRS              # currently same as J2000
LINEAR            # linear wcs as defined in file
AMPLIFIER         # mosaic coords of original file using
ATM/ATV
DETECTOR          # mosaic coords of original file using
DTM/DTV
```

Mosaic Images

While some coordinate systems are unique across mosaic images, others coordinate systems, such as `image`, or `physical`, are valid on a per segment basis. In this case, use `tile` to specify which header to use in all coordinate conversions. The default is the first header, or `tile 1`.

Example: `tile 2;fk5;point(100,100)`

Multiple WCS

If an image has multiple `wcs`'s defined, use `global wcs=wcs#` to specify which `wcs` to use for all `wcs` references. Valid values are `wcs`, `wcsa`, `wcsb`, `wcsc... wcz`. This is only valid as a global property.

Example: `global wcs=wcsa;linear;point(100,100) # point=diamond`

Specifying Positions and Sizes

The arguments to region shapes can be floats or integers describing positions and sizes. They can be specified as pure numbers or using explicit formatting directives:

position arguments

```
[num]                # context-dependent (see below)
[num]d               # degrees
[num]r               # radians
[num]p               # physical pixels
[num]i               # image pixels
[num]:[num]:[num]    # hms for 'odd' position arguments
[num]:[num]:[num]    # dms for 'even' position arguments
[num]h[num]m[num]s   # explicit hms
[num]d[num]m[num]s   # explicit dms
```

size arguments

```
[num]                # context-dependent (see below)
[num]"               # arc sec
[num]'               # arc min
[num]d               # degrees
[num]r               # radians
[num]p               # physical pixels
[num]i               # image pixels
```

When a "pure number" (i.e. one without a format directive such as 'd' for 'degrees') is specified, its interpretation depends on the context defined by the 'coordsys' keyword. In general, the rule is:

All pure numbers have implied units corresponding to the current coordinate system.

If no such system is explicitly specified, the default system is implicitly assumed to be PHYSICAL. In practice this means that for IMAGE and PHYSICAL systems, pure numbers are pixels. Otherwise, for all systems other than linear, pure numbers are degrees. For LINEAR systems, pure numbers are in the units of the linear system. This rule covers both positions and sizes. The input values to each shape can be specified in several coordinate systems including:

```
IMAGE                # pixel coords of current file
LINEAR               # linear wcs as defined in file

FK4, B1950           # various sky coordinate systems
FK5, J2000
GALACTIC
ECLIPTIC
ICRS
```

```

PHYSICAL                # pixel coords of original file using
LTM/LTV
AMPLIFIER                # mosaic coords of original file using
ATM/ATV
DETECTOR                 # mosaic coords of original file using
DTM/DTV

```

If no coordinate system is specified, PHYSICAL is assumed. PHYSICAL or a World Coordinate System such as J2000 is preferred and most general. The coordinate system specifier should appear at the beginning of the region description, on a separate line (in a file), or followed by a new-line or semicolon; e.g.,

```

image; circle 100 100 10
physical; ellipse 200 200 10 20
fk5; point 30 50

```

The use of celestial input units automatically implies WORLD coordinates of the reference image. Thus, if the world coordinate system of the reference image is J2000, then

```
circle 10:10:0 20:22:0 3'
```

is equivalent to:

```
j2000; circle 10:10:0 20:22:0 3'
```

Note that by using units as described above, you may mix coordinate systems within a region specifier; e.g.,

```
physical; circle 6500 9320 3'
```

External Region Files

DS9 can read and write a number of region file formats. Not all formats support all the functionality of DS9 regions. Therefore, the user may lose some information when writing and then reading back from a region file in a format other than DS9. On output, the regions File Format menu or the XPA regions point is used to specify the output coordinate system and format. On input, the menu or xpa point is used only for the X Y format. For all other formats, the input coordinate system is specified in the regions file itself.

Funtools

DS9 is fully compatible with Funtools, with the following notes:

Reading into Funtools from DS9:

- TEXT is ignored
- PROJECTION ignored
- RULER is ignored

- COMPASS is ignored
- All properties are ignored

Reading from Funtools into DS9:

- FIELD is ignored
- PIE is ignored

CIAO

- All point regions are translated as POINT
- BOX is translated as ROTBOX
- LINE ignored
- RULER is ignored
- COMPASS is ignored
- TEXT is ignored
- PROJECTION ignored
- ELLIPSE ANNULUS is ignored
- BOX ANNULUS is ignored
- PANDA is translated as PIE
- All properties are ignored

SAOimage

- All point regions are translated as POINT
- LINE is ignored
- TEXT is ignored
- PROJECTION ignored
- RULER is ignored
- COMPASS is ignored
- PANDA is ignored
- All properties are ignored

IRAF PROS

- All point regions are translated as POINT
- BOX is translated as ROTBOX
- LINE is ignored
- TEXT is ignored
- RULER is ignored
- COMPASS is ignored
- PROJECTION ignored
- PANDA is ignored
- All properties are ignored

FITS REGION Binary Table

- Read Only. DS9 currently can not write in this format.
- POINT is translated into BOX CIRCLE POINT
- ROTBOX is translated into BOX
- RECTANGLE is translated into BOX
- ROTRECTANGLE is translated into a BOX
- PIE is translated into PANDA
- The follow regions are not supported

- ELLIPTANNULUS
- SECTOR
- DIAMOND
- RHOMBUS
- ROTDIAMOND
- ROTRHOMBUS

X Y

This format consists of a number of coordinate pairs, one per line. The coordinate format for both input and output is specified via the Save Regions Parameters menu or XPA regions point. The first two coordinates are read, the rest of the line is ignored. The comment character '#' may be used at the beginning of line and the line is ignored. This format is very useful for reading in coordinates from other external analysis programs, such as IRAF.

Example:

```
# this is a comment
physical # this overrides the specified coordinate system
300 300
400 400 # this is a comment
500 500 this is ignored 550 700
```

- Save/List Regions-- all regions are output as a coordinate pairs.



Color

DS9 supports a number of color environments. Not all color environments, or visuals, are available on most machines. In fact, you may be restricted to one or two, based on the color graphics hardware your computer has. A color visual is composed of two parts, the color model and the bit depth. Pseudocolor uses a color lookup table to derive the correct color, Truecolor uses the value directly as a RGB triplet, to derive the correct color. The following is a list of the color visuals DS9 currently supports:

- pseudocolor, 8 bit
- truecolor, 8 bit
- truecolor, 15 bit
- truecolor, 16 bit
- truecolor, 24 bit

You can use the `xdpyinfo` command to see if one of these visual are available. NOTE: Linux Users-- if your desired visual is not available, use the Xconfigurator command (Red Hat) or similar command under other versions of linux, to configure your X window visuals.

When DS9 is invoked, by default, it will use the default visual. You can find out what the default visual is by using the `xdpyinfo` command. You can also force DS9 to use another visual by command line option. If you specify a visual, and it is not available, DS9 will exit with an error message.

```
$ds9 # default visual, default depth
$ds9 -visual pseudo # pseudocolor, default depth
$ds9 -visual pseudocolor # pseudocolor, default depth
$ds9 -visual pseudocolor8 # pseudocolor 8
$ds9 -visual true # truecolor, default depth
$ds9 -visual truecolor # truecolor, default depth
$ds9 -visual truecolor8 # truecolor 8
$ds9 -visual truecolor16 # truecolor 16
$ds9 -visual truecolor24 # truecolor 24
```



Printing

DS9 provides strong Postscript printing support. This is not a screen capture method, but a full level 1 and level 2 postscript driver. The postscript images generated are detailed and accurate as possible, given the resolution of the data, and the printing resolution.

Postscript Level

- Level 1-- The postscript generated consist of a color lookup table and image data, encoded in ASCIIHEX. All line graphics and text are postscript elements.
- Level 2-- The postscript generated consist of a color lookup table and image data, compressed with RLE, and encoded in ASCIIHEX85 . All line graphics and text are postscript elements.

Postscript Color Model

DS9 supports three color models for level 2 postscript. All three color models generate approximately the same size files.

- RGB
- CMYK
- Grayscale

Resolution

Most printers dither to achieve various levels of gray or color. So a 300 dpi printer's affective resolution may only be 53 dpi after dithering. This is fine for analysis and proofs. On the other hand, when generating images for publication, color separation is used to achieve the full resolution of the printer . A 150 dpi CMYK image will generate four 150 dpi images (one for each color). For example, ApJ requests submissions be postscript level 2, CMYK, 300 dpi. In reality, 150 dpi is more than adequate. See [Guidelines for Authors Submitting Electronic Art Files](#). In general, select the lowest resolution possible, as postscript file size grows by the square of the increase.

- 53 dpi-- use for 300 dpi printers
- 72 dpi-- use for 600 dpi printers
- 75 dpi-- use for 600 dpi printers
- 150 dpi-- use for publication
- 300 dpi-- use for publication
- 600 dpi-- use for publication



Contours

DS9 can create and display contours as an overlay on an image. The Display Contours menu is used to display contours. To create, copy, paste, and configure contours, use the Contour Parameters menu.

Contour Parameters

When creating a new contour, a dialog box appears, in which the user selects the number of contour levels, smoothness, and the distribution of the contours.

Contour Levels

Specifies the number of contour levels to be generated. A typical number is between 1 and 10. Note: large numbers of contours can take a long time to generate.

Contour Smoothness

Specifies how smooth the contours are. A smoothness level of 1 will evaluate the contour at each image pixel. A level of 2 will evaluate the contour at every other pixel. The larger the number, the quicker the contour will be generated, and the less detail will be available.

Contour Scale

Specifies the distribution of the contour levels. A linear distribution will have equal spacing between contours, Log and Ln will be weighted at one end. There are two ways to indicate the contour scale, Use Frame Scale (automatic) and manual

- Use Frame Scale
Use the color scale that the image is currently being displayed in. Therefore there is a one-to-one match between the image color scale distribution and the contour levels.
- Manual
Manually indicate upper and lower levels and a distribution for the contour levels. The contours generated will not be matched to the image color scale distribution.

Contour Method

There are two methods that are available to calculate the contour lines. The first, BLOCK, blocks down the image, by the smoothness factor, before contours are calculated. As a result, the larger the smoothness, the faster the result.

The second method, SMOOTH, smoothes the image before calculating contours. As a result, the larger the smoothness, the slower the result.



Coordinate Grids

DS9 can create and display coordinate grids as an overlay on an image. The Display Coordinate Grid Menu is used to display grids. A coordinate grid is composed of Grid Lines, Axes, Border, and Title. Axes include tickmarks, title, and numbers. The appearance of the coordinate grid is specified by parameters. These parameters may be configured via the Coordinate Grid Parameters dialog box. In addition to the axes titles and the grid title, the following menus are available.

File

Load and Save Coordinate Grid configurations.

Coordinate

Select the Coordinate system to be displayed.

View

Toggle the display of Grid lines, Title, Axes, Axes Numbers, and Border. Also specify Interior or Exterior Axes.

Color

Select the color of Grid lines, Title, Axes, Axes Title, Axes Numbers, Tickmarks, and Border.

Line

Select the line width and style (solid or dash) for the Grid lines, Axes, Tickmarks, and Border.

Font

Select the font family, font style, and font size for the grid Title, Axes Titles, and Axes Numbers.



Analysis

Each file type known to DS9 can have user-defined analysis commands associated with it. These analysis commands are defined at start-up time, or loaded by the user, by means of an ASCII analysis description file. The analysis commands are available for execution, either via the *Analysis Menu* or the XPA point *Analysis*. In addition, commands may be *bound* to events, such as keystrokes or mouse clicks. This type of command is called a bind command.

At startup, DS9 first searches for the analysis file, named *.ds9.ans*, in the local directory, then in the users home directory. A second analysis file to load at startup may be specified in the preferences (*Preferences : Analysis : Analysis File*). The user may also load or clear current analysis commands via command line options or the *Analysis menu*.

When activated, either from the menu, XPA, or bound event, an analysis command first is macro-expanded to fill in user-defined arguments and then is executed externally. Results may be displayed in a separate text window, plot window, or in a image frame.

[Syntax](#)

[Command Type](#)

[Macros](#)

[Help](#)

[Parameters](#)

[Hierarchical Menus](#)

[Sample](#)

Syntax

The analysis file that defines the known analysis commands consists of one or more file descriptors, each of which has the following format:

```
Menu label to be used
A space separated list of file templates
Command type [menu | bind <event>]
The command line for the analysis program
```

Task names may contain space characters. All lines may be indented. Also, the '#' character is a comment character. A separator can be inserted in the menu by specifying the following sequence '---'.

Example:

```
# this will insert a menu separator
---
```

Command Type

The third line indicates the type of command.

menu

A menu command creates an menu option under the *Analysis* menu option, and can be invoked by the user via the GUI or XPA. Example:

```
# Menu command example
My Analysis Task
*.fits
menu
$data | doit | $text
```

bind

A bind command is a command that is bound to an event. When the event occurs, the command is executed. Types of events available include all TK events, including all *keystrokes* and *mouse clicks*. If a command is bound to an event other than a *keystroke*, care must be taken to not to interfere with other internal DS9 events.

To bind to a key stroke, use the following command type:

```
bind <keystroke>
```

Example:

```
# Bind command example
Print coordinates
*.fits
bind x
echo "$x $y" | $text
```

web

A web command allows the user to invoke the internal web browser from the analysis menu.

Example:

```
# Web command example
HTTP based
*
web
http://hea-www.harvard.edu/RD/ds9/ref/index.html
File based
```

```
*
web
file:/home/joye/index.html
```

Macros

The following macros are macro-expanded to fill in user-defined arguments before the command is executed. Strings that contain `$<macroname>` that user does not want to be expanded may be escaped by using `$$<macroname>`. All strings that contain `$<string>` that are not a macro name will not be affected. For example:

```
echo "$$data $foo" | $text
```

will display a text dialog that contains "\$data \$foo"

\$bitpix

Substitute the bitpix of the data file in the command line.

Syntax:

```
$bitpix
```

Example:

```
echo "$bitpix" | $text
```

\$data

Data from the current frame becomes the input data to the command string. This data is in the form of a FITS image. This macro can only used at the beginning of the command string.

Syntax:

```
$data
```

Example:

```
$data | dosomething | $text
```

\$entry

Display an entry modal dialog. The returned string is substituted. If `cancel` is selected, the command line is not executed.

Syntax

```
$entry(<message>)
```

Example:

```
echo "$entry(Enter something here)" | $text
```

\$env

Substitute the value of a shell environment variable.

Syntax

```
$env(<shell variable>)
```

Example:

```
echo "$env(PATH)" | $text
```

\$filename

Substitute the filename of the data file in the command line. `$filename(root)` will return only the root filename.

Syntax:

```
$filename  
$filename(root)
```

Example:

```
dosomething $filename | $text  
dosomething $filename(root) | $text
```

\$filename[\$regions]

Combination of `$filename` and `$regions` macros. Generates a series of filenames, each with a region.

Syntax:

```
$filename[$regions]  
$filename[$regions(<options>)]
```

Example:

```
dosomething $filename[$regions] | $text
```

\$geturl

This macro differs from all other macros, including `$url`, in that no subprocess pipe is created. Only HTTP is supported. The contents of the url are retrieved and sent to `$text`, `$plot`, or `$image`. No other processing is allowed. The primary purpose of this macro is to support external analysis for the Windows platform, which has no subprocess support.

Syntax:

```
$geturl(http://<hostname>:<port>/<query>)
```

Example:

```
$geturl(http://foo.bar.edu/foo.html) | $text
```

\$image

The resulting image data is display in a DS9 frame. This macro should be the last macro of a command line. Optional parameter indicates if a new frame is created for the new data. The macro is removed from the command line before execution.

Syntax

```
$image  
$image([new|current])
```

Example:

```
doit | $image(new)
```

\$message

Display a message dialog box, with option buttons. After displaying the message, the macro is removed from the command line before execution. If `cancel` or `no` is selected, the command line is not executed.

Syntax

```
$message(<message>)  
$message([ok|okcancel|yesno],<message>)
```

Example:

```
$message(okcancel,This is a Message)| doit | $text
```

\$null

Expect no output or results from analysis task. Note: no error message will be returned if the analysis task fails to execute correctly.

Syntax

```
$null
```

Example:

```
echo "Hello, world" > foo | $null
```

\$plot

Display data in plot window. This macro should be the last macro of a command line. The data is read via `STDIN` and consist of a pair of coordinates, with option error values. (`xy`, `xyex`, `xyey`, `xyexey`) Default dimension is `xy`. The macro is removed from the command line before execution.

For `$plot(stdin)` only:

The title, x axis label, and y axis label are assumed to be on the first line of input, delimited with a new-line. However, if the data starts with `$BEGINTEXT`, all text between `$BEGINTEXT` and `$ENDTEXT` will be removed from the data and displayed in a separate text dialog window, with the remaining data, including the title, x axis label, and y axis label, will be displayed in a plot window. Furthermore, if the data contains the string `$ERROR`, an error is assumed to have occurred and a text dialog window is displayed only.

Syntax

```
$plot  
$plot(,,,)   
$plot(<title>,<x axis label>,<y axis  
label>,[xy|xyex|xyey|xyexey])  
$plot(stdin)
```

Example:

```
doit | $plot(This is aTitle,X Axis,Y Axis)  
doit | $plot(stdin)
```

\$regions

Substitute region definition in specified region format, coordinate system, and coordinate format. The default coord system is `physical`, default coordinate format `degrees`, and default region format `DS9`. Arguments may appear in any order, as long as they are separated by ',' and no spaces. If one or more properties are specified, only regions with all of the specified properties will be substituted.

Syntax:

```
$regions
$regions(<options>)
```

where options are one of the following:

```
regions format      = [ds9|ciao|saotng|saoimage|pros|xy]
property            = [include|exclude|source|background]
coordinate system   = [image|physical|detector|amplifier|wcs]
sky frame           = [fk4|fk5|icrs|galactic|ecliptic]
sky format          = [sexagesimal|degrees]
```

also, the old *SAOTNG* formats are also supported:

```
$regions_pixels
$regions_degrees
$regions_hms
$include_regions
$include_regions_pixels
$include_regions_degrees
$include_regions_hms
$exclude_regions
$exclude_regions_pixels
$exclude_regions_degrees
$exclude_regions_hms
```

Example:

```
dosomething $regions | $text
dosomething $regions(pros) | $text
dosomething $regions(source,wcs,fk5) | $text
dosomething
$regions(saotng,background,exclude,ecliptic,sexagesimal) | $text
```

\$text

Display text in a text dialog window. This macro should be the last macro of a command line. To display text from only `STDOUT` use '|' as the pipe command. To display text from both `STDOUT` and `STDERR`, use '|&' as the pipe command. No parameters are required. The macro is removed from the command line before execution.

Syntax

```
$text
```

Example:

```
doit | $text # stdout
doit |& $text # stdout and stderr
```

\$url

URLs are processed and stored in a temporary file. Only HTTP and anonymous FTP are supported.

Syntax:

```
$url(http://<hostname>:<port>/<query>)
$url(ftp://<hostname>/<filename>)
```

Example:

```
$url(http://legacy.gsfc.nasa.gov/rosat/data/p000s26b.img.Z) |
uncompress | $image
```

```
$url(ftp://legacy.gsfc.nasa.gov/rosat/data/hri/images/rh100193_img.fits)
| $image
```

\$x

\$y

Substitute coordinates of an bind event. When a bind event is triggered, the x,y coordinates of the mouse of the particular data file are returned. The default coordinate system is *physical*. This macro is only available for bind commands.

Syntax:

```
$x
$x(<coordinate system>,<format>)
$y
$y(<coordinate system>,<format>)
```

where:

```
coordinate system = [image|physical|detector|amplifier|wcs]
sky frame         = [fk4|fk5|icrs|galactic|ecliptic]
sky format        = [hms|sexagesimal|degrees]
```

Example:

```
echo $x(fk5,sexagesimal) $y(fk5,sexagesimal) | $text
```

\$xdim

\$ydim

Substitute the width or height of the data file in the command line.

Syntax:

```
$xdim
$ydim
```

Example:

```
echo "$xdim $ydim" | $text
```

\$xpa

Returns the xpa access point name.

Syntax:

```
$xpa
```

Example:

```
echo '$xpa' | $text
```

\$xpamethod

Returns the xpa method.

Syntax:

```
$xpamethod
```

Example:

```
echo '$xpamethod' | $text
```

Help

The user may define his own HELP message. This message will be available to the user as a menu item. An optional label maybe specified. The default label is Help. When invoked, an text dialog window will appear, containing the message. Multiple HELP items maybe defined within a menu or across hierarchical menus.

Example:

```
help Main Help
A help message may contain
multiple lines of description of the tasks
in the menu or menus
endhelp
```

Parameters

The user may define his own macros or parameters to be evaluated before the command line is executed. To do this, the user defines a param segment that is referenced in the command line. The param definition has the follow format:

```
param <name>
<variable> <entry | checkbox | menu> <title> <default> <{comment}>
...
endparam
```

or

```
param <name>
@<iraf param filename>
end
```

The definition either consisted of a number of variables, one per row, or the name of a IRAF style parameter file. DS9 will look for the IRAF parameter file in:

```
./<filename>  
$UPARM/<filename>  
$HOME/iraf/<filename>
```

Example:

```
param foobar  
var1 entry {Variable 1} default {this is a entry}  
var2 checkbox {Variable 2} 1 {this is a checkbox}  
var3 menu {Variable 3} AAA|BBB|CCC {this is a menu}  
endparam
```

To use parameters, specify the param name at the beginning of your command line:

```
Parameter Test  
*  
menu  
$param(foobar); echo "$var1 $var2 $var3" | $text
```

When the menu item is selected, the user will be presented with a dialog box that contains *entry*, *checkbox*, or *menu* choices for each variable specified. If the user clicks ok, the values are substituted in the command line before execution.

Hierarchical Menus

The user may define hierarchical menus. Use this to organized crowded menus. To do this, frame menu entries with `hmenu <label>` and `endhmenu`. Hierarchical menu labels may contain spaces. Multiple levels maybe implimented.

Example:

```
hmenu Stuff  
  hello  
  *  
  menu  
  echo "Hello" | $text  
  
  world  
  *  
  menu  
  echo "World" | $text  
  
hmenu More Stuff  
  hello world  
  *  
  menu
```

```

        echo "Hello World" | $text
    endhmenu
endhmenu

```

Will create an hierarchical menu with two members, hello and world.

Sample

```

#
# Analysis command descriptions:
#     menu label
#     file templates
#     menu/bind
#     analysis command line
param foo
    var1 entry entry 40 {this is a entry}
    var2 checkbox checkbox 1 {this is a checkbox}
    var3 menu menu AAA|BBB|CCC {this is a menu}
endparam
param bar
    @analysis.par
endparam
param foobar
    @tvdisply.par
endparam
param ltc
    bins entry "Enter number of [t1:t2:]bins" 0 "('0' for default
number of bins)"
endparam
# Help Main Help
help Main Help
These menus contain a test for each possible feature
supported by the ds9 (blank line above)
endhelp
---
hmenu Test Web
    help Web Help
    Help for web features
    endhelp
    Web Test url
    *
    web
    http://hea-www.harvard.edu/RD/ds9/
    Web Test file
    *
    web
    file:/home/joye/saods9/ds9/tests/hv.html

```

```

endhmenu
hmenu Test Basics
  help Basic Help
  Help for basic features
endhelp
---
Test escape char # this is a comment
*
menu
echo "this is not a macro $$xpa" | $text
Test pass thru # this is a comment
*
menu
echo "this is not a macro $foo" | $text
Test $xpa # this is a comment
*
menu
echo $xpa | $text
Test $xpa_method#this is also a comment
*
menu
echo $xpa_method | $text
Test $filename
*.fits
menu
echo $filename | $text
Test $filename(root)
*.fits
menu
echo $filename(root) | $text
Test $xdim $ydim $bitpix
*.fits
menu
echo "$xdim $ydim $bitpix" | $text
Test $env
*
menu
echo $env(PATH) | $text
endhmenu
hmenu Test Regions
  help Regions Help
  Help for regions features
endhelp
---
Test $regions
*.fits
menu

```

```

    echo "$regions ds9_s:$regions(ds9,source,image)
ciao_b:$regions(ciao,background)
saotng_i:$regions(saotng,include,wcs,fk5)
pros_e:$regions(pros,exclude,wcs,fk5,sexagesimal)
xy_be:$regions(xy,background,exclude,wcs,fk4,hms)" | $text
    Test $regions wcs
    *.fits
    menu
    echo "$regions(ds9,wcs) $regions(ds9,wcs,fk5,sexagesimal)
$regions(ds9,wcsa) " | $text
    Test $include_regions_pixels
    *.fits
    menu
    echo "ds9_s: $source_regions ds9_b: $background_regions_pixels
ds9_i: $include_regions_degrees ds9_e: $exclude_regions_hms" | $text
    Test $filename $regions
    *.fits
    menu
    echo "$filename[$regions]" | $text
    Test $filename $regions()
    *.fits
    menu
    echo "$filename[$regions()]" | $text
endhmenu
hmenu Test Output
    help Output Help
    Help for output features
endhelp
---
Test $null
*
menu
echo "This is Text" > /dev/null | $null
Test $text
*
menu
echo "This is Text" | $text
Test $text stderr
*
menu
ls foofoofoo | $text
Test $plot
*
menu
cat xy.dat | $plot
Test $plot(title,x,y,xyey)
*

```

```

menu
cat xye.dat | $plot(Title,X Axis,Y Axis,xyey)
Test $plot(title,x,y,xyexey)
*
menu
cat xye.dat | $plot(Title,X Axis,Y Axis,xyexey)
Test $plot(title,x,y,4)
*
menu
cat xyey.dat | $plot(Title,X Axis,Y Axis,4)
Test $plot(title,x,y,5)
*
menu
cat xyeye.dat | $plot(Title,X Axis,Y Axis,5)
Test $plot(stdin)
*
menu
cat xye.stdin.dat | $plot(stdin)
Test $plot(stdin) text
*
menu
cat xye.stdin.text.dat | $plot(stdin)
Test $plot(stdin) error
*
menu
cat xy.stdin.error.dat | $plot(stdin)
Test $data
*.fits
menu
$data | $image(new)
Test $image
*
menu
cat img16.fits | $image
endhmenu
hmenu Test Dialogs
help Dialogs Help
Help for dialog features
endhelp
---
Test $message(message)
*
menu
$message(ok,This is a Message) | echo "hello" | $text
Test $message(ok,message)
*
menu

```

```

    $message(ok,This is a Message) | echo "World" | $text
    Test $entry(message)
    *
    menu
    echo "$entry(Enter Something)" | $text
endhmenu
hmenu Test Params
    help Param Help
    Help for param features
    endhelp
    ---
    Test $param
    *
    menu
    $param(foo); echo "$var1 $var2 $var3" | $text
    Test $param @file
    *
    menu
    $param(bar); echo "$var1 $var2 $var3" | $text
endhmenu
hmenu Test Network
    help Network Help
    Help for network features
    endhelp
    ---
    Test $url(http://)
    *
    menu

$url(http://legacy.gsfc.nasa.gov/FTP/rosat/data/cdrom/vol1/IMAGES/00h/p000s26b.img.Z)
| gunzip | $image
    Test $url(ftp://)
    *
    menu

$url(ftp://legacy.gsfc.nasa.gov/rosat/data/hri/images/fits/rh100193_img.fits)
| $image
    Test $geturl $text
    *
    menu

$geturl(http://hea-www.harvard.edu/RD/saord-cgi/funtools?funcnts+$filename+$regions(source,,)+$regions(background,,))|$text
    Test $geturl $plotstd
    *
    menu
    $param(ltc);
    $geturl(http://hea-www.harvard.edu/RD/saord-cgi/funtools?funhist_plot+$filename[$regions]+time+$bins)|$plot(stdin)
endhmenu

```

```

hmenu Test Other
  help Other Help
  Help for other features
endhelp
---
Test $param @tvdisply
*
menu
  $param(foobar); echo "$frame $erase" | $text
hmenu Test MultiLevel
  test
  *
  menu
    echo "Hello World" | $text
endhmenu
endhmenu
$x $y
*.fits
bind x
echo "$x $y" | $text
$x(fk5,hms) $y(fk5,hms)
*.fits
bind y
echo "$x(fk5,hms) $y(fk5,hms)" | $text
$x(wcs,fk5,hms) $y(wcs,fk5,hms)
*.fits
bind z
echo "$x(wcs,fk5,hms) $y(wcs,fk5,hms)" | $text

```



File Formats

Information on the following file formats are available:

[FITS](#)

[FITS Multiple Extension Data Cube](#)

[FITS Mosaic](#)

[FITS RGB](#)

[Split FITS](#)

[Data Array](#)

[External Format Support](#)

[External Analysis Support](#)

[Region Files](#)

[Color Lookup Table](#)

[WCS](#)

[Preference File](#)

[Startup File](#)

[TCL](#)

FITS

DS9 supports FITS images, FITS bin tables and FITS ascii tables. See [FITS Standard](#) for more information. At load time, the user may provide just a filename or a filename along with FITS extension name or number, and columns names for FITS bin tables. FITS extension names and columns names are case insensitive.

Syntax:

```
filename
filename[ext]
filename[bin]
filename[filter]
filename[ext][bin]
filename[ext][filter]
filename[bin][filter]
filename[ext][bin][filter]
filename[ext, bin]
filename[bin, filter]
filename[ext, bin, filter]
```

where ext

name

number

and bin

```

bin=colx,coly # bin counts
bin colx,coly # bin counts
bin=colx,coly,colz # bin on colz
bin colx,coly,colz # bin on colz
bin=colz # bin cols 'x', 'y', and colz
key=colx,coly
binkey=colx,coly
and filter (see Introduction to Filtering for more information)
any filter expression

```

Example:

```

$ds9 foo.fits # default load
$ds9 foo.fits[1] # load first extension, image or bin table
$ds9 foo.fits[BCKGRD] # load extension named 'BCKGRD'
$ds9 foo.fits[bin=detx,dety] # load FITS bin table, bin on detx,dety
$ds9 foo.fits[2][bin=rawx,rawy] # load FITS bin table, ext 2, cols
rawx,rawy
$ds9 foo.fits[bg_events,bin=rawx,rawy] # load FITS bin table
$ds9 foo.fits[bin=x,y,pha] # load FITS bin table, bin on x,y,pi
$ds9 foo.fits[bin=pi] # load FITS bin table, bin on x,y,pi
$ds9 foo.fits[events][pha>5,pi<2] # load extension 'events' and
filter

```

The shell environment variable `DS9_BINKEY` may be used to specify default bin cols for FITS bin tables. Example:

```

$export DS9_BINKEY='[bin=rawx,rawy]'
$ds9 foo.fits # load FITS bin table, bin on rawx, rawy

```

The following algorithm is used to locate and to load the FITS image if no additional information is provide:

```

Look for FITS image in primary HDU.
If no image is found, examine each extension HDU
  If image, load
  if bin table, load if the following is true
    extension name is EVENTS or STDEVT or RAYEVENT
    column names X and Y are present
If DS9 traverses the entire FITS file without satisfying one of
the above, an error is generated.

```

FITS keyword inheritance is supported. All valid FITS BITPIX values are supported, along with -16, for UNSIGNED SHORT. The following FITS keywords are supported:

- OBJECT
- BSCALE / BZERO
- BLANK
- DATASEC

- LTV / LTM for physical coords
- DTV / DTM for detector coords
- ATV / ATM for amplifier coords
- WCS keywords
- WCS# keywords

FITS Multiple Extension Data Cube

A FITS Multiple Extension Data Cube is a FITS file with one or more extensions, that is to be displayed as a data cube. Each image does not have to be the same size, however, only the coordinate systems from the first extension will be used for contours and grids.

Example:

```
$ds9 -medatacube foo.fits # load multiple extension fits file as
data cube
```

FITS Mosaic

A FITS mosaic image may exist as a series of FITS files, or as one FITS file with many extensions. A FITS mosaic may be loaded all at one time, or by the segment. Once loaded, the multiple FITS images are treated as one FITS image.

DS9 supports three forms of mosaics:

IRAF	contains the DETSEC and DETSIZE keywords. See NOAO IRAF Mosaic Data Structures
WCS	each FITS image or FITS bin table must contain a valid equatorial wcs.
HST WFPC2	valid HST WFPC2 data cube, consisting of 4 planes, along with a fits ascii table containing wcs information.

In General, DS9 by default will first try to load as an IRAF mosaic. If the IRAF keywords are not present, DS9 will then try to load as a WCS mosaic.. This behavior can be modified to force either IRAF or WCS from the preference menu.

Example:

```
$ds9 -mosaicimageiraf foo.fits # load mosaic iraf from one fits file
with multiple exts
$ds9 -mosaiciraf foo.fits bar.fits wow.fits # load mosaic iraf from
3 files
$ds9 -mosaicimagewcs foo.fits # load mosaic wcs from one fits file
with multiple exts
$ds9 -mosaicimagenext wcs foo.fits # load mosaic wcs from one fits
file with multiple exts
$ds9 -mosaicwcs foo.fits bar.fits wow.fits # load mosaic wcs from 3
files
```

```
$ds9 -mosaicimagewfpc2 bar.fits # load wfpc2 mosaic
$ds9 -mosaic foo.fits bar.fits wow.fits # load mosaic (iraf or wcs)
from 3 files
```

FITS RGB

A FITS RGB image may exist as three of FITS images, one FITS file with three extensions, or as a FITS 3D Data cube, with three slices, each representing the red, green, and blue channel. A FITS RGB image may be loaded all at one time, or by the channel. Once loaded, the multiple FITS images are treated as one FITS image.

Example:

```
$ds9 -rgbimage rgb.fits # load rgb image consisting of one fits file
with 3 image exts
$ds9 -rgbcube cube.fits # load rgb image consisting of one fits data
cube
$ds9 -rgb -red foo.fits -green bar.fits -blue wow.fits # rgb image
from 3 fits images
```

Split FITS

A split fits is a valid fits file in which two files contain the header and data segments.

Data Array

Raw data arrays are supported. To load an array, the user must provide the dimensions, pixel depth, and optional header size and architecture type.

Syntax:

```
filename[options]
```

options are:

```
xdim=value
ydim=value
zdim=value # default is a depth of 1
dim=value
dims=value
bitpix=[8|16|-16|32|-32|-64]
skip=value # must be even, most must be factor of 4
arch=[bigendian|littleendian]
```

Example:

```
$ds9 -array bar.arr[xdim=512,ydim=512,zdim=1,bitpix=16] # load
512x512 short
$ds9 -array bar.arr[dim=256,bitpix=-32,skip=4] # load 256x256 float
with 4 byte head
$ds9 -array bar.arr[dim=512,bitpix=32,arch=littleendian] # load
512x512 long, intel
```

or alternate format:

```
filename[array(<type><dim><:skip><endian>)]  
type:
```

```
'b' 8-bit unsigned char  
's' 16-bit short int  
'u' 16-bit unsigned short int  
'i' 32-bit int  
'r' 32-bit float  
'f' 32-bit float  
'd' 64-bit float
```

dim:

```
int      # x,y dim  
int.int  # x,y dim  
int.int.int # x,y,z dim
```

skip:

```
int      # number of bytes to skip
```

endian:

```
'l' little endian  
'b' big endian
```

Example:

```
$ds9 -array bar.arr[array(s512)] # load 512x512 short  
$ds9 -array bar.arr[array(r256:4)] # load 256x256 float with 4 byte  
head  
$ds9 -array bar.arr[array(i512l)] # load 512x512 long, intel
```

The shell environment variable `DS9_ARRAY` may be used to specify default array parameters.

Example:

```
$export DS9_ARRAY='[dim=256,bitpix=-32]'  
$ds9 -array foo.arr # load 256x256 float
```

External File Support

DS9 supports external file formats via an ASCII description file. When loading a file into DS9, these descriptions are referenced for instructions for loading the file, based on the file extension. If found, the command is executed and the result, a FITS image or FITS Binary Table, is read into DS9 via `stdin`.

At start-up, DS9 first searches for the ASCII file, named *.ds9.fil*, in the local directory, then in the users home directory.

The file command first is macro-expanded to fill in user-defined arguments and then is executed externally.

The ASCII file that defines the known image files consists of one or more file descriptors, each of which has the following format:

- Help description
- A space-separated list of templates
- A space-separated list of file types (not currently used)
- The command line for the loading this file type

Note that blank lines separate the file descriptions and should not be used as part of a description. Also, the '#' character is a comment character.

The following macros are supported:

Macro	Description
filename	insert current full file name

For Example:

```
# File access descriptions:
#     help explanation
#     file template
#     file type
#     access command
```

```
IRAF IMH files
*.imh
IMH
i2f -s $filename
```

External Analysis Support

For more information about external analysis support files, see [Analysis](#).

Region Files

DS9 can read and write a number of region file formats. See [Regions](#) documentation for more information.

- [DS9](#)
- [FUNTools](#)
- [Ciao](#)
- [SAOimage](#)
- [IRAF PROS](#)
- [FITS REGION Binary Table](#)
- [X Y](#)

Color Lookup Table

DS9 has a number of default colormaps available to the user. DS9 also supports reading and writing color lookup table formats from the following programs:

- SAOimage
- SAOtng
- XImtool
- Skycat

DS9 uses the file extension to determine the color table format:

Ext	Format
.lut	XImtool, SAOtng
.lasc	Skycat
.sao	DS9, SAOimage
any other	DS9

WCS

A new WCS specification can be loaded and used by the current image regardless of the WCS that was contained in the image file. WCS specification can be sent to DS9 as an ASCII file via XPA. The format of the specification is a set of valid FITS keywords that describe a WCS.

Example:

```
CRPIX1 = 257.75
CRPIX2 = 258.93
CRVAL1 = -201.94541667302
CRVAL2 = -47.45444
CDELTA1 = -2.1277777E-4
CDELTA2 = 2.1277777E-4
CTYPE1 = 'RA---TAN'
CTYPE2 = 'DEC--TAN'
```

Note that the WCS definitions can contain standard FITS 80 character WCS card images, as shown above, or free-form name/value pairs without the intervening "=" sign:

```
CRPIX1 257.75
CRPIX2 258.93
CRVAL1 -201.94541667302
CRVAL2 -47.45444
CDELTA1 -2.1277777E-4
CDELTA2 2.1277777E-4
CTYPE1 'RA---TAN'
CTYPE2 'DEC--TAN'
```

Preference File

A preference file is a valid tcl script generated by DS9 to save the current preference items. See [Preferences](#) for more information.

Startup File

If a startup file is available, `~/ds9.ini` or `.ds9.ini` for windows, it is sourced as the last step in initialization.

TCL

TCL/TK script file. Users may customize the appearance and enhance the capabilities of DS9 by sourcing their own TCL scripts.



Preferences

Allows the user to customize the appearance and behavior of the GUI . Please note: some preferences take affect immediately, while others require DS9 to be restarted. Use `Save Preferences` to store changes and `Clear Preferences` to restore default settings.

User preferences are stored in the file `~/ .ds9 .prf` for non-windows platforms and `.ds9 .prf` for windows. DO NOT EDIT this file, since it will be deleted or overwritten by DS9.

For non-windows platforms, system wide preferences maybe implemented by copying a `.ds9 .prf` to `/usr/local/lib/ds9.prf` or `/opt/local/lib/ds9.prf` . At startup time, if no user preference file is located in the user's home directory, DS9 will search for these system wide preferences.

Users may have several different preference files. DS9 looks for a preference file with its own name. By default, if the application is named `ds9`, it will look for `~/ .ds9 .prf` . However, if the DS9 application is named `f00`, then DS9 will look for `~/ .f00 .prf` . In this manner, the user can have several predefined preference files that are activated by invoking DS9 with a different application names.

[File Menu](#)

[Edit Menu](#)

[Frame Menu](#)

[Bin Menu](#)

[Zoom Menu](#)

[Scale Menu](#)

[Color Menu](#)

[Region Menu](#)

[WCS Menu](#)

[Analysis Menu](#)

[Background Color](#)

[Blank/NaN Color](#)

[View](#)

[Info Panel](#)

[Default Menu Font](#)

[Mosaics](#)

[Panner](#)

[Magnifier](#)

[Truecolor Colorbar](#)

[Dialog Box](#)

Pan
Print Coordinates
Examine
Misc
HTTP

Save Preferences
Clear Preferences

File Menu

Configure default Print and Page Setup Dialog.

Edit Menu

Set the default mouse mode.

Frame Menu

- Tile
Configure the tiling pattern to be used while in Tile mode.
- Blink Interval
Set the blink interval while blinking between frames.
Sets the size of the display frame, within DS9.
- Auto Size
Sets the frame size to be as big as possible, given the current window size.
- 256x256
Sets the frame size to be 256x256, centered in the window.
- 512x512
Sets the frame size to be 512x512, centered in the window.
- 1024x1024
Sets the frame size to be 1024x1024, centered in the window.

Bin Menu

Set the default bin function, factor, and buffer size.

Zoom Menu

Set the default Zoom factor, Orientation, and Rotation, along with WCS Alignment.

Scale Menu

Set the default scale, limits, and to use DATASEC if available.

Color Menu

Set the default Colormap.

Region Menu

Specify default region parameters such as shape, color, width, properties, fonts, radius and width and height.

WCS Menu

Specify default wcs parameters. The default wcs, sky frame, and sky format will be used as default settings for the information panel, panner wcs compass, wcs alignment, grid.

Analysis Menu

- Horizontal Graph
 - Grid
Show/Hide the grid on the Horizontal graph.
 - Linear
Configure the graph for linear data.
 - Log
Configure the graph for log data. IMPORTANT: your data min value must be ≥ 1 for it to display properly.
- Vertical Graph
 - Grid
Show/Hide the grid on the Vertical graph.
 - Linear
Configure the graph for linear data.
 - Log
Configure the graph for log data. IMPORTANT: your data min value must be ≥ 1 for it to display properly.
- Name Server
Configure the default Name server. Used by both the Analysis Menu Items Name Server Dialog and DSS Server Dialog.
- DSS Server
Configure the default DSS server. Used by the Analysis Menu item DSS Server Dialog.
- Archive Servers
Add up to 4 additional URL's to the Analysis Archive Server Menu. Changes take affect immediately
- Preload Analysis File
The name of up to 4 Analysis Files to be preloaded at startup.
- Analysis Command Log. List final command line before execution.

- Analysis Plot
Default settings for all Analysis Plots

Background Color

Set the default color to be used for the image background.

Blank/INF/NaN Color

Set the default color to be used with a image value of BLANK, INF or NaN is encountered.

View

Show/Hide display components

- Information Panel
- Buttons
- Panner
- Magnifier
- Colorbar

Info Panel

Controls what information is displayed in the Information Control Panel.

- Filename
Show/Hide base name of the current FITS image.
- Object
Show/Hide the contents of the FITS keyword OBJECT, if present.
- Min Max
Show/Hide the min and max values for the current FITS image.
- Frame Info
Show/Hide the frame number, zoom, and angle of the current frame.
- WCS
Show/Hide WCS coordinate values, for the current FITS image, if present.
- Detector (optional)
Show/Hide Detector coordinate values, for the current FITS image. Requires the DTV/DTM keywords be present in the header. See [NOAO IRAF Mosaic Data Structures](#) for more information.
- Amplifier (optional)
Show/Hide Amplifier coordinate values, for the current FITS image. Requires the ATV/ATM keywords be present in the header. See [NOAO IRAF Mosaic Data Structures](#) for more information.
- Physical
Show/Hide Physical or CCD coordinate values, for the current FITS image. Requires the LTV/LTM keywords be present in the header. See [NOAO IRAF Mosaic Data Structures](#) for more information.

- Image
Show/Hide Image or Logical coordinate values, for the current FITS image.

Default Menu Font

Select the default font for menus and buttons. Please note that you must save the preferences and restart ds9 for the default font changes to take affect.

Mosaics

Configure default behavior for loading IRAF or WCS mosaics

- Auto Detect- first attempt to load as IRAF, then WCS.
- Force IRAF- attempt to load only as IRAF mosaic.
- Force WCS- attempt to load only WCS.
- Fast Mosaic Render - enable optimized mosaic render. Only valid for IRAF mosaics. May give incorrect results if the display areas of the mosaic segments overlap.

Panner

- Orientation Compass
Show/Hide the image orientation compass.
- WCS Compass
Show/Hide/Set coordinate system for the WCS compass.
- WCS Compass Tangent Point
Display the WCS compass at the tangent point or the center of the panner.

Magnifier

- Graphics
Show/Hide regions and crosshair in the magnifier window. This is not available for the 'Fast' display algorithm.
- Cursor
Show/Hide the cursor in the center of the magnifier window. This cursor displays the size and orientation of one pixel, given the current zoom, angle and orientation of the frame.
- Magnification
Current magnification factor of the image rendered into the magnifier.

Truecolor Colorbar

To optimize adjusting the colorbar contrast and bias on slower truecolor machines, the user may select a subsection of the image to be manipulated.

Dialog Box

- Motif
All Standard Dialog boxes will be in Motif style.
- Windows
All Standard Dialog boxes will be in Microsoft Windows style.
- Blue Plate Special
Custom dialog box which allows filename wild card characters.

Pan

Configure the interactive pan method.

- Click to Center
Center image at click point. Same method as SAOimage.
- Drag to Center
Click, drag, and release to pan the image in the current frame
- Pan then Zoom
Pan to center, next 2 clicks zoom at center, click 4 resets zoom to 1. This behavior emulates the middle button mouse behavior of XIMTOOL.

Print Coordinates

Specify which coordinates are printed when the 'C' key is pressed. The format of WCS coordinates are also specified. Also, the value of the closest pixel may also be printed.

Examine

Configure the Examine function. While in examine mode, the user may indicate that all new examine operations go to one frame, or create a new frame each time. Also, the user may indicate the scale factor to be used in the examine frame. If the base frame is an image, the zoom factor is changed, if the base frame is a FITS Binary Table, the block factor is changed.

Misc

- Specify which directory is to be used for temporary files. The user must have write access to the directory. Under Unix, the default is `/tmp`. Under other platforms, the default is the current directory.
- Select WCS Projection algorithms: FITS WCS or AIPS. The default is FITS.
- Use Cursors: change the mouse cursor over regions to indicate edit options
- Autoload Regions: When loading a FITS Image or Bin Table from a file, scan for FITS Regions Table and load if present.
- Show Amplifier/Detector in Menus: Enable / Disable Amplifier and Detector coordinate system menu options. Does not take affect til next startup.
- Initialize XPA: Enable/Disable XPA access at startup. Does not take affect til next startup.

HTTP

Set HTTP Proxy and authentication configuration.

Save Preferences

Saves the current values of all preference items in the preference file.

Clear Preferences

Clears all current preferences. Actually deletes the preferences file, if present. The next time DS9 is started, default settings are used.



IRAF Support

DS9 is a fully functional IRAF image display server. IRAF uses the IIS protocol to communicate with a valid image display server, such as DS9, ximtool, saomage, and saotng. With DS9, no special scripts are needed. If you have one of the above currently working, DS9 works *right out of the box*. And DS9 now supports IRAF's new IIS image display protocol that supports up to 16 display frames.

All native DS9 functions may be used with images load with IRAF display except for the `Scale` menu items. Values displayed may be the true values, if a linear scale is specified with the `display` command. Otherwise, the value is a scaled value. DS9 supports IRAF in all display visuals including `Truecolor`. Support full postscript printing of images loaded from IRAF is provided.

Command Line Arguments

As with *ximtool*, the follow command line arguments may be used to specify the communication parameters:

- `-fifo`
- `-fifo only`
- `-inet only`
- `-port`
- `-port only`
- `-unix`
- `-unix only`

The default parameters are:

- `fifo /dev/imt1`
- `port 5137`
- `unix /tmp/.IMT%d`

Configuration

An *IRAF* image server uses a configuration file to specify the number of available buffers and their sizes. What actually passes from IRAF is not the buffer size, but an index number into this file.

So when an image server starts (DS9), it will attempt to locate this file as `$HOME/.imtoolrc` and `/usr/local/lib/imtoolrc`. If not found, it will look for shell environment variables `IMTOOLRC` and `imtoolrc`, that contains the name of the configuration file.

If no configuration file is found, DS9 will assume the following default configuration:

```
1  2  512  512  # imt1|imt512
2  2  800  800  # imt2|imt800
3  2 1024 1024  # imt3|imt1024
4  1 1600 1600  # imt4|imt1600
5  1 2048 2048  # imt5|imt2048
6  1 4096 4096  # imt6|imt4096
7  1 8192 8192  # imt7|imt8192
8  1 1024 4096  # imt8|imt1x4
9  2 1144  880  # imt9|imtfs full screen (1152x900 minus frame)
10 2 1144  764 # imt10|imtfs35 full screen at 35mm film aspect
ratio
11 2  128  128  # imt11|imt128
12 2  256  256  # imt12|imt256
13 2  128 1056  # imt13|imttall128 tall & narrow for spectro.
14 2  256 1056  # imt14|imttall256 tall & wider for spectro.
15 2 1056  128  # imt15|imtwide128 wide & thin for spectro.
16 2 1056  256  # imt16|imtwide256 wide & fatter for spectro.
17 2 1008  648  # imt17|imtssy Solitaire fmt w/ imtool border
18 2 1024  680  # imt18|imtssn Solitaire fmt w/out imtool
border
19 1 4096 1024  # imt19|imt4x1
```

If on the other hand, IRAF assumes a different buffer size, the image will appear corrupted and DS9 may issue a number of error messages.

Another problem is that this file must be in sync with `dev$graphcap`. If your system administrator has made changes to `graphcap`, they must also be implemented in `imtoolrc`.

Here is a note from NOAO:

```
The messages means that there is no /usr/local/lib/imtoolrc file
on the machine. This is created as a symlink to dev$imtoolrc by
the
iraf install script but only if the /usr/local/lib dir already
exists on the
machine. The fix is the create the dir and rerun the install
script or
else make the link by hand.  Users can also just copy
dev$imtoolrc
to $HOME/.imtoolrc and restart the server to also workaround it.
Note
that an existing .imtoolrc might define old frame buffer configs
which
might confuse things, so if the system file exists check for a
private
copy screwing things up.
```

Windows DS9 and IRAF

To direct image output from IRAF to DS9 running under windows, use the `IMTDEV` environment variable. For example, if the windows machine is named 'foo.bar.edu', define `IMTDEV` to the follow value before entering IRAF.

```
$ setenv IMTDEV inet:5137:foo.bar.edu
$ cl
cl> display dev$pix
```

Scale Menu Disabled

When you display an image from *IRAF* into DS9, *IRAF* actually does the color scale distribution. In `Display`, use the `ztrans` and `z1,z2` parameters to set the upper/lower bounds and distribution. You can also use the `zscale` parameter to auto determine `z1 , z2` . Here are the `DISPLAY` parameters in question:

```
ztrans=[linear|log|none|user]
z1=min
z2=max
zscale=[yes|no]
```

What actually is sent from *IRAF* to DS9 is one byte per pixel, values 0-200, which already has applied both the upper and lower clipping bounds and the distribution. So this is why, the `SCALE` menu is disabled in DS9 when it receives a image from *IRAF*.

MSCRED/MSCZERO

DS9 now supports IRAF's new IIS image display protocol. However, there is one minor problem with the **mscred** task **msczero**. Before using **msczero**, issue the following command in the `cl`:

```
cl> set disable_wcs_maps=""
cl> flpr
```

IMEXAMINE

Due to the unique relationship between DS9 and IRAF, if you use the **imexamine** task, you can take advantage of a special feature of DS9. Instead of loading the image from IRAF with the **display** task, load the image directly into DS9. Then, from the `cl` prompt, invoke **imexamine** without a filename. IRAF will ask DS9 for the current filename and use it for analysis. This approach provides several advantages over previous methods. First, it will work with compound fits images such as mosaics, data cubes, and rgb images. Second, the image displays includes true image data and WCS information, not the approximated data from IRAF.