Table of Contents

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.

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
View
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. Mblockquotetiple 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 Data Cube - Load entire FITS image composed of multiple extensions as a data cube. The standard file dialog is presented.
Open Multi Ext Multi Frames - Load entire FITS image composed of multiple extensions into multiple frames. 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`
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 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 Image ...

Save current image as fits, jpeg, tiff, png, ppm, or mpeg file. Only the visible portion of the image is saved, along with all graphs including regions, catalogs, contours, and grids. If the image is a data cube, saving as a mpeg movie will generate a mpeg-1 movie, one frame per slice of data.

Save Frame as Fits...

Save the current frame as a fits image. This differs from above in that all image data is saved, not just the visible data. Also, no rotation, zoom, or panning is applied. The main purpose of this feature is to allow a user to load a FITS binary table, and save it as a FITS image.

Save Frames as MPEG...

Save all frames as a MPEG-1 movie. Each frame is cycled thru and the visible image is saved as one frame of the movie, along with all graphics including regions, catalogs, grids, and contours.

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.

Open TCL Console

Open a TCL console in a separate window.

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 behavior of the mouse when moved, clicked, or dragged over various sections of the DS9 display.

None 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 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 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 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 an 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.

Catalog Mode

Select / Deselect catalog regions and highlite catalog table entry if available.

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. Therefor, 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.

View Menu

Specify DS9 layout

```
Horizontal
Vertical
```

Show/Hide DS9 display components

```
Information Panel
Panner
Magnifier
Buttons
Colorbar
Horizontal Graph
Vertical Graph
```

Show/Hide Information panel components

```
Filename
Object
Min/Max data values
Low/High data values
Frame Information
WCS Coordinates
Image Coordinates
Physical Coordinate
```

Configure Colorbar

```
Horizontal
Vertical
Show Numerics
Specify Font
```

Frame Menu

New Frame

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

New Frame RGB

Creates a new rgb frame. Each frame has an 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.

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.

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.

Move Frame

Change the order of display

Goto Frame

Goto a particular frame

Data Cube

Displays Fits Data Cube dialog.

RGB

Displays RGB Frame dialog.

Frame Parameters

Specify Tile, Blink, and display size parameters

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 Block In/Out

Increase or decrease Bin block factor

Bin to Fit Frame

Bin to Fit Frame will calculate a bin block 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.

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.

Center Frame

Will pan to the center of the current frame.

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 bye the `DETSIZE` FITS keyword.

Align to WCS

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

Zoom In/Out

Zoom in/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.

Zoom Factor

Set current zoom factor.

Orientation

Set current orientation.

Rotation

Set current rotation.

Pan Zoom Rotate Parameters

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 determined these values, based on the Min Max Options.

99.5% to 90%

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

ZMax

Set the lower limit base on the IRAF ZScale algorithm and the upper limit on the maximum data value. ZScale parameters may be configured via the preference menu.

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 minimum and maximum values of the image. A selection of AUTO will use SAMPLE for all FITS Mosaics and SCAN for all other files.

ZScale

Set zscale parameters

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 /bias 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.

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.

Composite Region

Create a new composite region from all selected regions or dissolve an existing composite region into the original components. Composite regions can not be nested.

Instrument FOV

Specify an instrument field of view to be created and displayed.

Template

Load or Save an template regions file.

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.

Centroid

Centroid the selected regions

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.

Invert Selection

Invert or toggle the current selection.

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.

File Format

Specify the regions file format. Supported formats are DS9 / Funtools, Ciao, SAOtng, SAOimage, IRAF pros and X Y coordinate pairs. See Regions for more information.

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.

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

Region Parameters

Configure region parameters such as Show, Show Text, Auto Centroid and Centroiding Parameters.

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

```
  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.

Pixel Table

Display table of pixel values

Mask Parameters

Display the Mask dialog box. Load Masks, and specify mask transparency.

Contours

Toggle the display of contours using the current contour parameters. Create, copy, paste, and configure contours from image data. For more information, see Contours.

Coordinate Grid

Toggle the display of a coordinate grid using the current grid parameters. Display a non-modal dialog box to allow modification of coordinate grid parameters. For more information, see Coordinate Grids.

Smoothing

Toggle smoothing using current smooth parameters. Support for smoothing images, binary tables, data cubes, and mosaics.

Name Resolution

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

Image Servers

Enables access to image servers. Internet access must be available for this feature to work.

Archives

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.

Catalogs

Support for loading, displaying, filtering, and saving catalogs. Most major catalogs can be retrieved from online servers.

Virtual Observatory

Enables access to supported virtual observatories from DS9. Internet access must be available for this feature to work. For more information, see Virtual Observatory Reference

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.

Catalog Tool

Bring up catalog 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.

Virtual Observatory Reference

Summary

Use the Web proxy connection if your firewall does not allow your computer to connect directly to external computers. In this case, you also must use DS9's internal browser.

Details

When you click on one of the Virtual Observatory servers in the VO list, DS9 will attempt to connect to that server and (if the internal Web display is enabled) display its Web page. The square box to the left of the server name turns yellow while the connection is being established and then green to signal success.

A direct connection is fast and flexible. Among other things, it allows you to perform analysis on your own local data (the VO server will retrieve the image from DS9) and also allows you to use an external browser to load images.

Some system managers configure their firewall explicitly to prevent computers in their care from making a direct connection to an external host. Instead, they only allow external access through a Web proxy
server (such as SOCKS). If you are using a computer behind a restricted firewall of this sort, then DS9 will not be able to connect directly to a VO server. The yellow box will not turn green and eventually DS9 will display an error message.

In this case, you can choose to have DS9 communicate with the VO servers through your Web proxy server. DS9 will use your proxy to send its commands and retrieve its data and analysis results, rather than doing this directly. Note that the following restrictions apply:

The transfer of data is slower.
You must use the internal Web browser for loading images, etc.
You cannot perform analysis on local data.
There is a (large but finite) restriction on the number of annuli, and number of polygon points you can specify in a region, as well as the total number of regions allowed.

If your computer and firewall have been configured to require use of a Web proxy server, you will have to tell DS9 about this server. Click the **Configure Web Proxy** button and type the relevant information into the boxes. (Your systems administrator will be able to tell you the details.) At this point, you should be able to connect to a VO server successfully. Please let us know if you have problems!

A final note: you may, of course, choose to use the Web proxy even if your computer and firewall are configured to allow direct connections. In this case, there is no need to configure the proxy server.

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 internal web browser. Can be used to download the latest DS9 version.

Catalogs

DS9 provides full support for loading, displaying, filtering, and saving catalogs. DS9 allows you to overlay symbols from multiple catalogs on the current image.

Local and on-line catalog access is supported. Most major catalogs can be retrieved from online servers. Both the CDS and SDSS catalog servers are now supported. Local catalog files in starbase (rdb) or CSV (with or without header) are supported.

On-line catalogs are available via services provided by the VizieR catalog access tool, CDS, Strasbourg, France (VizieR is a joint effort of the Centre de Données Astronomiques de Strasbourg and ESA-ESRIN Information Systems Division) and by the Sloan Digital Sky Survey.

A selection of popular catalogs is provided in the Analysis menu. In addition, you can search for other catalogs based on title, keywords, mission, wavelength, and object type.

When a catalog is overlayed on an image, each displayed catalog symbol consists of a shape, color, and text. An advanced symbol editor is available that allows you to specify the shape, size, color, and text of each symbol, based on catalog column values. These symbol expressions can be saved for future use.

Along with the overlay display, a catalog list is provided in a separate window. It displays the column values for each catalog object. The catalog list can be sorted and filtered, and the catalog display will be automatically updated. Advanced filtering options are available. Catalogs can be loaded and saved as local files in ASCII Starbase format. Each catalog contains header information which can be displayed. The list can be printed separately from the image.

An interactive connection between the displayed catalog symbols and the catalog list is provided. When you select one or more rows within the catalog list, the corresponding symbols are highlighted on the image display. Conversely, selecting multiple symbols on the image display will highlight the corresponding rows within the catalog list. Catalog symbols can be converted to regions for use with analysis tasks.

Filter Option

The catalog list can be sorted and filtered, and the catalog display will be automatically updated. A filter is conditional expression, when evaluated for each row of the catalog, if true, the row is displayed, and if false, the row is not displayed. The conditional expression can be any valid TCL expression. The value of a column may be indicated with `$<column name>`.

```
$_RAJ2000>180. && $_RAJ2000<270.
$Jmag>11
log($Kmag*10)<.3
```

Advanced Symbol Editor

An advanced symbol editor is available that allows you to specify the shape, size, color, and text of each symbol, based on catalog column values. For each row of the catalog, one or more conditional expressions are evaluated. For the first expression to evaluate true, a given symbol is displayed, with the specified shape, color, size and text properties. As with the filter, the value of a particular column can be indicated as `$<column name>.`

For the condition entry, the expression you type in is automatically evaluated via TCL `expr` after macro expansion.

```
1                              # always
0                              # never
true                           # always
false                          # never
$Jmag>2                        # conditional
sin($Jmag)>.5                  # conditional
[string equal $Class SNR]      # conditional
[regexp {*SNR*} $Class]        # conditional
```

For the size, size2, and angle entries, the expression you type in is also automatically evaluated via TCL `expr` after macro expansion.

```
2               # value of ’2’ is used
$Jmag           # value of column Jmag is used
$Jmag/2.        # value of column Jmag div 2 is used
(4+2)/3         # value of ’2’ is used
```

For the text portion, this is not true. It is assumed to be text, unless you explicitly use an `expr` operator.

```
foo             # will put ’foo’ above the symbol
$Jmag           # will put the value of column Jmag above the
symbol
(4+2)/3         # will put the text ’(4+2)/3’ above the symbol
[expr (4+2)/3]  # will put the text ’2’ above the symbol
[expr $Jmag/2.] # will take the value of Jmag and div by 2
```

And finally, one special case for shape = text and text = empty. In this case, the row number is displayed.

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 EDIT 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-‘`	Rotate thur all open windows

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, base on the color graphics hardware your computer has. A color visual is composed of two parts, the color model and the bit depth. Pseudo color uses a color lookup table to derive the correct color, True color uses the value directly as a RGB triplet, to derive the correct color. The follow is a list of the color visuals DS9 currently supports:

```
pseudo color, 8 bit
true color, 8 bit
true color, 15 bit
true color, 16 bit
true color, 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 Xconfigarator 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         # pseudo color, default depth
$ds9 -visual pseudocolor    # pseudo color, default depth
$ds9 -visual pseudocolor8   # pseudo color 8
$ds9 -visual true           # true color, default depth
$ds9 -visual truecolor      # true color, default depth
$ds9 -visual truecolor8     # true color 8
$ds9 -visual truecolor16    # true color 16
$ds9 -visual truecolor24    # true color 24
```

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` and the scale changed to `zscale`. Next, `bar.fits` is loaded and the colormap for `bar.fits` is changed to `I8.`

`2mass`
`about`
`align`
`analysis`
`array`
`bin`
`blink`
`blue`
`catalog`
`cd`
`cmap`
`colorbar`
`console`
`contour`
`crosshair`
`cursor`
`datacube`
`dsssao`
`dsseso`
`dssstsci`
`exit`
`fifo`
`fifo_only`
`file`
`first`
`fits`
`sfits`
`frame`
`geometry`
`green`

grid
header
height
help
histequ
iconify
inet_only
invert
iis
language
linear
lock
log
lower
magnifier
mask
match
medatacube
minmax
mode
mosaicimage
mosaic
multiframe
smosaic
msg
nameserver
nvss
orient
pagesetup
pan
pixeltable
plot
prefs
preserve
psprint
print
private
port
port_only
pow
quit
raise
regions
rgb
rgbarray
rgbcube
srgbcube

rgbimage
red
rotate
samp
saveimage
savefits
savempeg
scale
shm
single
skyview
sleep
smooth
squared
sqrt
source
tcl
tile
title
unix
unix_only
update
url
version
view
visual
vo
wcs
web
width
xpa
zmax
zscale
zoom

2mass

Support for 2MASS Digital Sky Survey.

```
Syntax:
-2mass [<object>]
       [name <object>]
       [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
       [size <width> <height> degrees|arcmin|arcsec]
       [save yes|no]
       [frame new|current]
       [update frame|crosshair]
       [survey j|h|k]
       [open|close]

Example:
$ds9 -2mass m31
$ds9 -2mass name m31
$ds9 -2mass coord 00:42:44.404 +41:16:08.78 sexagesimal
$ds9 -2mass size 60 60 arcmin
$ds9 -2mass save yes
$ds9 -2mass frame current
$ds9 -2mass update frame
$ds9 -2mass survey j
$ds9 -2mass open
$ds9 -2mass close
```

about

Get DS9 credits.

```
Syntax:
-about

Example:
$ds9 -about
```

align

Controls the World Coordinate System alignment for the current frame.

```
Syntax:
-align [yes|no]

Example:
$ds9 -align yes
```

analysis

Control external analysis tasks. Tasks are numbered as they are loaded, starting with 0. Can also be used to display a message and display text in the text dialog window.

```
Syntax:
-analysis [<task number>]
          [<filename>]
          [task <task number>|<task name>]
          [load <filename>]
          [clear]
          [clear][load <filename>]
          [message ok|okcancel|yesno <message>]
          [entry <message>]
          [text]
```

```
Example:
$ds9 -analysis 0 # invoke first analysis task
$ds9 -analysis task 0
$ds9 -analysis task foobar
$ds9 -analysis task "{foo bar}"
$ds9 -analysis my.ans
$ds9 -analysis load my.ans
$ds9 -analysis clear
$ds9 -analysis clear load my.ans
$ds9 -analysis message '{This is a message}'
$ds9 -analysis message okcancel '{This is a message}'
$ds9 -analysis text '{This is text}';
```

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>]
     [about center]
     [buffersize <value>]
     [cols <x> <y>]
     [colsz <x> <y> <z>]
     [factor <value> [<value>]]
     [depth <value>]
     [filter <string>]
     [function average|sum]
     [to fit]
     [open|close]

Example:
$ds9 -bin about 4096 4096
$ds9 -bin about center
$ds9 -bin buffersize 512
$ds9 -bin cols detx dety
$ds9 -bin colsz detx dety time
$ds9 -bin factor 4
$ds9 -bin factor 4 2
$ds9 -bin depth 10
$ds9 -bin filter '{pha > 5}'
$ds9 -bin filter ''
$ds9 -bin function sum
$ds9 -bin to fit
$ds9 -bin open
$ds9 -bin close
```

blink

Blink mode parameters. Interval is in seconds.

```
Syntax:
-blink []
       [yes|no]
       [interval <value>]

Example:
$ds9 -blink
$ds9 -blink yes
$ds9 -blink interval 1
```

blue

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

```
Syntax:
-blue

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

catalog
cat

Support for catalogs.

```
Syntax:
-catalog []
         [ned|simbad|denis|skybot]

[ascss|cmc|gsc1|gsc2|gsc3|ac|nomad|ppmx|sao|sdss5|sdss6|tycho|ua2|ub1|ucac2]
         [2mass|iras]
         [csc|xmm|rosat]
         [first|nvss]
         [chandralog|cfhtlog|esolog|stlog|xmmlog]
         [cds <catalogname>]
         [cds <catalogid>]
         [load <filename>]
         [load [xml|sb|tsv] <filename>]
         [<catname>] [samp]
         [<catname>] [samp broadcast]
         [<catname>] [samp send <application>]
         [<catname>] [name <object>]
         [<catname>] [coordinate <ra> <dec> <coordsys>]
         [<catname>] [size <width> <height> degrees|arcmin|arcsec]
         [<catname>] [save <filename>]
         [<catname>] [save [xml|sb|tsv] <filename>]
         [<catname>] [filter <string>]
         [<catname>] [filter load <filename>]
         [<catname>] [clear]
         [<catname>] [retrieve]
         [<catname>] [cancel]
         [<catname>] [print]
         [<catname>] [close]
         [<catname>] [server
cds|sao|cadc|adac|iucaa|bejing|cambridge|ukirt]
         [<catname>] [symbol [#]
condition|shape|color|font|fontsize|fontweight|fontslant <value>]
         [<catname>] [symbol [#] text|size|size2|units|angle
<value>]
         [<catname>] [symbol shape {circle point}|{box
point}|{diamond point}|
                     {cross point}|{x point}|{arrow
point}|{boxcircle point}|
                     circle|ellipse|box|text]
         [<catname>] [symbol add| [#] remove]
         [<catname>] [symbol save|load <filename>]
         [<catname>] [sort <columnname> incr|decr]
         [<catname>] [maxrows <number>]
         [<catname>] [allrows]
         [<catname>] [allcols]
         [<catname>] [ra <columnname>]
         [<catname>] [dec <columnname>]
         [<catname>] [system <coordsys>]
         [<catname>] [sky <skyframe>]
         [<catname>] [show|hide]
         [<catname>] [edit yes|no]
         [<catname>] [panto yes|no]

Example:
$ds9 -catalog
$ds9 -catalog 2mass
$ds9 -catalog cds 2mass
$ds9 -catalog cds "I/252"
$ds9 -catalog load foo.cat
$ds9 -catalog load xml foo.xml
$ds9 -catalog samp broadcast
$ds9 -catalog samp send aladin
$ds9 -catalog cat2mass symbol color red
$ds9 -catalog name m51
$ds9 -catalog coordinate 202.48 47.21 fk5
$ds9 -catalog size 22 22 arcmin
$ds9 -catalog save bar.cat
$ds9 -catalog save xml bar.xml
$ds9 -catalog filter "\$Jmag>10"
$ds9 -catalog filter load foo.flt
$ds9 -catalog clear
$ds9 -catalog retrieve
$ds9 -catalog cancel
$ds9 -catalog print
$ds9 -catalog close
$ds9 -catalog server sao
$ds9 -catalog symbol condition "\$Jmag>15"
$ds9 -catalog symbol 2 shape "boxcircle point"
$ds9 -catalog symbol color red
$ds9 -catalog symbol font times
$ds9 -catalog symbol fontsize 14
$ds9 -catalog symbol fontweight bold
$ds9 -catalog symbol fontslant italic
$ds9 -catalog symbol add
$ds9 -catalog symbol 2 remove
$ds9 -catalog symbol load foo.sym
$ds9 -catalog symbol save bar.sym
$ds9 -catalog sort "Jmag" incr
$ds9 -catalog maxrows 2000
$ds9 -catalog allrows
$ds9 -catalog allcols
$ds9 -catalog ra RA
$ds9 -catalog dec DEC
$ds9 -catalog system wcs
```

```
$ds9 -catalog sky fk5
$ds9 -catalog show
$ds9 -catalog hide
$ds9 -catalog edit yes
$ds9 -catalog panto no
```

cd

Sets the current working directory.

```
Syntax:
cd [<directory>]

Example:
$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>]
      [open|close]

Example:
$ds9 -cmap Heat
$ds9 -cmap file foo.sao
$ds9 -cmap invert yes
$ds9 -cmap value 5 .5
$ds9 -cmap open
$ds9 -cmap close
```

colorbar

Controls colorbar parameters.

```
Syntax:
-colorbar [yes|no]
          [horizontal|vertical]
          [orientation horizontal|vertical]
          [numerics yes|no]
          [space value|distance]
          [font times|helvetica|courier]
          [fontsize <value>]
          [fontweight normal|bold]
          [fontslant roman|italic]
          [size]
          [ticks]

Example:
$ds9 -colorbar yes
$ds9 -colorbar vertical
$ds9 -colorbar orientation vertical
$ds9 -colorbar numerics yes
$ds9 -colorbar space value
$ds9 -colorbar font times
$ds9 -colorbar fontsize 14
$ds9 -colorbar fontweight bold
$ds9 -colorbar fontslant italic
$ds9 -colorbar size 20
$ds9 -colorbar ticks 11
```

console

Display tcl console window.

```
Syntax:
-console

Example:
$ds9 -console
```

contour

Controls contours in the current frame.

```
Syntax:
-contour []
         [yes|no]
         [clear]
         [generate]
         [load <filename> <coordsys> <skyframe> <color> <width> 
yes|no]
         [save <filename> <coordsys> <skyframe>]
         [convert]
         [loadlevels <filename>]
         [savelevels <filename>]
         [copy]
         [paste <coordsys> <color> <width> yes|no]
         [color <color>]
         [width <width>]
         [dash yes|no]
         [smooth <smooth>]
         [method block|smooth]
         [nlevels <number of levels>]
         [scale linear|log|pow|squared|sqrt|histequ]
         [log exp <value>]
         [mode minmax|<value>|zscale|zmax]
         [limits <min> <max>]
         [levels <value value value...>]
         [open|close]

Example:
$ds9 -contour
$ds9 -contour yes
$ds9 -contour generate
$ds9 -contour clear
$ds9 -contour load ds9.con wcs fk5 yellow 2 no
$ds9 -contour load ds9.con wcs fk5 red 2 yes
$ds9 -contour save ds9.con wcs fk5
$ds9 -contour convert
$ds9 -contour loadlevels ds9.lev
$ds9 -contour savelevels ds9.lev
$ds9 -contour copy
$ds9 -contour paste wcs red 2 no
$ds9 -contour color yellow
$ds9 -contour width 2
$ds9 -contour dash yes
$ds9 -contour smooth 5
$ds9 -contour method smooth
$ds9 -contour nlevels 10
$ds9 -contour scale sqrt
$ds9 -contour log exp 1000
$ds9 -contour mode zscale
$ds9 -contour limits 1 100
$ds9 -contour levels "1 10 100 1000"
$ds9 -contour open
$ds9 -contour close
```

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 <coordsys> [<skyframe>][<skyformat>]]

Example:
$ds9 -crosshair 100 100 physical # set crosshair in physical
```

```
$ds9 -crosshair 345 58.8 wcs fk5 # set crosshair in wcs coords
$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:
$ds9 -cursor 10 10
```

datacube

Controls FITS datacube.

```
Syntax:
-datacube [play|stop|next|prev|first|last]
          [#]
          [interval #]
          [axis #]
          [open|close]

Example:
$ds9 -datacube play
$ds9 -datacube last
$ds9 -datacube 3
$ds9 -datacube interval 2
$ds9 -datacube axis 3
$ds9 -datacube open
$ds9 -datacube close
```

dsssao

Support for Digital Sky Survey at SAO.

```
Syntax:
-dsssao [<object>]
        [name <object>]
        [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
        [size <width> <height> degrees|arcmin|arcsec]
        [save yes|no]
        [frame new|current]
        [update frame|crosshair]
        [open|close]
```

```
Example:
$ds9 -dsssao m31
$ds9 -dsssao name m31
$ds9 -dsssao coord 00:42:44.404 +41:16:08.78 sexagesimal
$ds9 -dsssao size 60 60 arcmin
$ds9 -dsssao save yes
$ds9 -dsssao frame current
$ds9 -dsssao update frame
$ds9 -dsssao open
$ds9 -dsssao close
```

dsseso

Support for Digital Sky Survey at ESO.

```
Syntax:
-dsseso [<object>]
        [name <object>]
        [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
        [size <width> <height> degrees|arcmin|arcsec]
        [save yes|no]
        [frame new|current]
        [update frame|crosshair]
        [survey DSS1|DSS2-red|DSS2-blue|DSS2-infrared]
        [open|close]

Example:
$ds9 -dsseso m31
$ds9 -dsseso name m31
$ds9 -dsseso coord 00:42:44.404 +41:16:08.78 sexagesimal
$ds9 -dsseso size 60 60 arcmin
$ds9 -dsseso save yes
$ds9 -dsseso frame current
$ds9 -dsseso update frame
$ds9 -dsseso survey DSS2-red
$ds9 -dsseso open
$ds9 -dsseso close
```

dssstsci

Support for Digital Sky Survey at STSCI.

```
Syntax:
-dssstsci [<object>]
          [name <object>]
          [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
          [size <width> <height> degrees|arcmin|arcsec]
          [save yes|no]
          [frame new|current]
          [update frame|crosshair]
          [survey poss2ukstu_red|poss2ukstu_ir|poss2ukstu_blue]
          [survey poss1_blue|poss1_red]
          [survey all|quickv|phase2_gsc2|phase2_gsc1]
          [open|close]

Example:
$ds9 -dssstsci m31
$ds9 -dssstsci name m31
$ds9 -dssstsci coord 00:42:44.404 +41:16:08.78 sexagesimal
$ds9 -dssstsci size 60 60 arcmin
$ds9 -dssstsci save yes
$ds9 -dssstsci frame current
$ds9 -dssstsci update frame
$ds9 -dssstsci survey all
$ds9 -dssstsci open
$ds9 -dssstsci close
```

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
```

file

Load FITS file. Note, this the default mode. Use this command to reset after using -mosaicimage, -mosaic, -array, and -mask.

```
Syntax:
-file <filename>

Example:
$ds9 -file foo.fits
```

first

Support for VLA First Sky Survey.

```
Syntax:
-first [<object>]
       [name <object>]
       [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
       [size <width> <height> degrees|arcmin|arcsec]
       [save yes|no]
       [frame new|current]
       [update frame|crosshair]
       [open|close]

Example:
$ds9 -first m31
$ds9 -first name m31
$ds9 -first coord 00:42:44.404 +41:16:08.78 sexagesimal
$ds9 -first size 60 60 arcmin
$ds9 -first save yes
$ds9 -first frame current
$ds9 -first update frame
$ds9 -first open
$ds9 -first close
```

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]]
       [move first]
       [move back]
       [move forward]
       [move last]
       [first]
       [prev]
       [next]
       [last]
       [frameno #]
       [#]

Example:
$ds9 -frame center # center current frame
$ds9 -frame center 1 # center 'Frame1'
$ds9 -frame center all # center all frames
$ds9 -frame clear # clear current frame
$ds9 -frame new # create new frame
$ds9 -frame new rgb # create new rgb frame
$ds9 -frame delete # delete current frame
$ds9 -frame reset # reset current frame
$ds9 -frame refresh # refresh current frame
```

```
$ds9 -frame hide # hide current frame
$ds9 -frame show 1 # show frame ’Frame1’
$ds9 -frame move first # move frame to first in order
$ds9 -frame move back # move frame back in order
$ds9 -frame move forward # move frame forward in order
$ds9 -frame move last # move frame to last in order
$ds9 -frame first # goto first frame
$ds9 -frame prev # goto prev frame
$ds9 -frame next # goto next frame
$ds9 -frame last # goto last frame
$ds9 -frame frameno 4 # goto frame ’Frame4’, create if needed
$ds9 -frame 3 # goto frame ’Frame3’, create if needed
```

geometry

Define the initial window geometry. This includes all of the ds9 window, not just the image space. 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. For grid numeric format syntax, click here.

```
Syntax:
-grid []
      [yes|no]
      [type analysis|publication]
      [system <coordsys>]
      [sky <skyframe>]
      [skyformat <skyformat>]
      [grid yes|no]
      [grid color <color>]
      [grid width <value>]
```

```
[grid style 0|1]
[grid gap1 <value>]
[grid gap2 <value>]
[axes yes|no]
[axes color <color>]
[axes width <value>]
[axes style 0|1]
[axes type interior|exterior]
[format1 <format>]
[format2 <format>]
[tick yes|no]
[tick color <color>]
[tick width <value>]
[tick style 0|1]
[border yes|no]
[border color <color>]
[border width <value>]
[border style 0|1]
[numlab yes|no]
[numlab font times|helvetica|courier]
[numlab fontsize <value>]
[numlab fontweight normal|bold]
[numlab fontslant roman|italic]
[numlab color <color>]
[numlab gap1 <value>]
[numlab gap2 <value>]
[numlab type interior|exterior]
[numlab vertical yes|no]
[title yes|no]
[title text <text>]
[title def yes|no]
[title gap <value>]
[title font times|helvetica|courier]
[title fontsize <value>]
[title fontweight normal|bold]
[title fontslant roman|italic]
[title color <color>]
[textlab yes|no]
[textlab text1 <text>]
[textlab def1 yes|no]
[textlab gap1 <value>]
[textlab text2 <text>]
[textlab def2 yes|no]
[textlab gap2 <value>]
[textlab font times|helvetica|courier]
[textlab fontsize <value>]
[textlab fontweight normal|bold]
```

```
      [textlab fontslant roman|italic]
      [textlab color <color>]
      [reset]
      [load <filename>]
      [save <filename>]
      [open|close]

Example:
$ds9 -grid
$ds9 -grid yes
$ds9 -grid type analysis
$ds9 -grid system wcs
$ds9 -grid sky fk5
$ds9 -grid skyformat degrees
$ds9 -grid grid yes
$ds9 -grid grid color red
$ds9 -grid grid width 2
$ds9 -grid grid style 1
$ds9 -grid grid gap1 10
$ds9 -grid grid gap2 10
$ds9 -grid axes yes
$ds9 -grid axes color red
$ds9 -grid axes width 2
$ds9 -grid axes style 1
$ds9 -grid axes type exterior
$ds9 -grid format1 d.2
$ds9 -grid format2 d.2
$ds9 -grid tick yes
$ds9 -grid tick color red
$ds9 -grid tick width 2
$ds9 -grid tick style 1
$ds9 -grid border yes
$ds9 -grid border color red
$ds9 -grid border width 2
$ds9 -grid border style 1
$ds9 -grid numlab yes
$ds9 -grid numlab font courier
$ds9 -grid numlab fontsize 12
$ds9 -grid numlab fontweight bold
$ds9 -grid numlab fontslant italic
$ds9 -grid numlab color red
$ds9 -grid numlab gap1 10
$ds9 -grid numlab gap2 10
$ds9 -grid numlab type exterior
$ds9 -grid numlab vertical yes
$ds9 -grid title yes
$ds9 -grid title text {Hello World}
```

```
$ds9 -grid title def yes
$ds9 -grid title gap 10
$ds9 -grid title font courier
$ds9 -grid title fontsize 12
$ds9 -grid title fontweight bold
$ds9 -grid title fontslant italic
$ds9 -grid title color red
$ds9 -grid textlab yes
$ds9 -grid textlab text1 {Hello World}
$ds9 -grid textlab def1 yes
$ds9 -grid textlab gap1 10
$ds9 -grid textlab text2 {Hello World}
$ds9 -grid textlab def2 yes
$ds9 -grid textlab gap2 10
$ds9 -grid textlab font courier
$ds9 -grid textlab fontsize 12
$ds9 -grid textlab fontweight bold
$ds9 -grid textlab fontslant italic
$ds9 -grid textlab color red
$ds9 -grid reset
$ds9 -grid load foo.grd
$ds9 -grid save foo.grd
$ds9 -grid open
$ds9 -grid close
```

header

Display current fits header dialog. Optional extension number maybe specified.

```
Syntax:
-header [<value>]
        [close [<value>]]

Example:
$ds9 -header
$ds9 -header 2
$ds9 -header close
```

height

Set the height of the image display window. Use the geometry command to set the overall width and height of the ds9 window.

```
Syntax:
-height [<value>]

Example:
$ds9 -height 512
```

help

Display help information. To maintain backward compatability, -help will display a brief help message and exit. --help will display all command line options within the built-in help facility.

```
Syntax:
-help # Display brief help message and exit.
--help # Display command line options within help facility.
-? # Display command line options within help facility.

Example:
$ds9 -help # Display brief help message and exit.
$ds9 --help # Display command line options within help facility
$ds9 -? # Display command line options within help facility.
```

histequ

Select histogram equalization scale for current frame.

```
Syntax:
-histequ

Example:
$ds9 -histequ
```

iconify

Toggles iconification.

```
Syntax:
-iconify []
         [yes|no]

Example:
$ds9 -iconify
$ds9 -iconify yes
```

invert

Invert Colormap.

```
Syntax:
-invert

Example:
$ds9 -invert
```

iis

Set IIS Filename. Optional mosaic number maybe supplied.

```
Syntax:
-iis [filename <filename> [#]]

Example:
$ds9 -iis filename foo.fits
$ds9 -iis filename bar.fits 4
```

language

Select current language.

```
Syntax:
-language [locale|da|de|es|en|fr|ja|pt]

Example:
$ds9 -language fr
```

linear

Select Linear scale for current frame.

```
Syntax:
-linear

Example:
$ds9 -linear
```

lock

Lock frames.

```
Syntax:
lock [crosshair none|wcs|wcsa...wcsz|physical|image]

Example:
$ds9 -lock crosshair wcs
```

log

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

```
Syntax:
-log

Example:
```

```
$ds9 -log
```

lower

Lower in the window stacking order.

```
Syntax:
-lower

Example:
$ds9 -lower
```

magnifier

Controls the magnifier settings.

```
Syntax:
magnifier [color <color>]
          [zoom <value>]
          [cursor yes|no]
          [region yes|no]

Example:
$ds9 -magnifier color yellow
$ds9 -magnifier zoom 2
$ds9 -magnifier cursor no
$ds9 -magnifier region no
```

mask

Controls mask parameters.

```
Syntax:
-mask [color <color>]
      [mark 1|0]
      [transparency <value>]
      [clear]
      [open|close]

Example:
$ds9 -mask color red
$ds9 -mask mark 0
$ds9 -mask transparency 50
$ds9 -mask clear
$ds9 -mask open
$ds9 -mask close
```

match

Match all other frames to the current frame.

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

Example:
$ds9 -match frames wcs
$ds9 -match colorbars
$ds9 -match scales
$ds9 -match bin
```

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. DATAMINIRAFMIN 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 [auto|scan|sample|datamin|irafmin]
        [mode auto|scan|sample|datamin|irafmin]
        [interval <value>]

Example:
$ds9 -minmax scan
$ds9 -minmax mode scan
$ds9 -minmax interval 10
```

mode

Select the current mode.

```
Syntax:
-mode
[none|pointer|crosshair|colorbar|pan|zoom|rotate|catalog|examine]

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
```

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
```

msg

Specify a directory of translation tables to be loaded.

```
Syntax:
-msg <directory>

Example:
$ds9 -msg $HOME/msgs
```

multiframe

Load multiple extension FITS file into multiple frames. NOTE: multiframe can not be used for files that have been loaded via stdin.

```
Syntax:
-multiframe <filename>

Example:
$ds9 -multiframe foo.fits
```

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]
            [pan]
            [crosshair]
            [open|close]

Example:
$ds9 -nameserver m31
$ds9 -nameserver name m31
$ds9 -nameserver server ned-sao
$ds9 -nameserver skyformat sexagesimal
$ds9 -nameserver pan
$ds9 -nameserver crosshair
$ds9 -nameserver open
$ds9 -nameserver close
```

nvss

Support for NRAO VLA Sky Survey.

```
Syntax:
-nvss [<object>]
```

```
      [name <object>]
      [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
      [size <width> <height> degrees|arcmin|arcsec]
      [save yes|no]
      [frame new|current]
      [update frame|crosshair]
      [open|close]

Example:
$ds9 -nvss m31
$ds9 -nvss name m31
$ds9 -nvss coord 00:42:44.404 +41:16:08.78 sexagesimal
$ds9 -nvss size 60 60 arcmin
$ds9 -nvss save yes
$ds9 -nvss frame current
$ds9 -nvss update frame
$ds9 -nvss open
$ds9 -nvess close
```

orient

Controls the orientation of the current frame.

```
Syntax:
-orient [none|x|y|xy]
        [open|close]

Example:
$ds9 -orient xy
$ds9 -orient open
$ds9 -orient close
```

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 <coordsys> [<skyframe>][<skyformat>]
     [to x y <coordsys> [<skyframe>][<skyformat>]
     [open|close]

Example:
$ds9 -pan 200 200 image
$ds9 -pan to 400 400 physical
$ds9 -pan to 13:29:55 47:11:50 wcs fk5
$ds9 -pan open
$ds9 -pan close
```

pixeltable

Display/Hide the pixel table.

```
Syntax:
-pixeltable []
            [yes|open]
            [no|close]

Example:
$ds9 -pixeltable
$ds9 -pixeltable yes
$ds9 -pixeltable open
$ds9 -pixeltable close
```

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>]]
      [new [name <plotname>] <title> <xaxis label> <yaxis label>
 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>] [print destination 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|courier]
      [<plotname>] [font numbers|labels|title size <value>]
      [<plotname>] [font numbers|labels|title weight normal|bold]
      [<plotname>] [font nubmers|labels|title slant roman|italic]
# edit current dataset
-plot [<plotname>] [dataset #]
      [<plotname>] [view discrete|linear|step|quadratic|error
yes|no]
      [<plotname>] [color discrete|linear|step|quadratic|error
<color>]
      [<plotname>] [line discrete circle|diamond|plus|cross]
      [<plotname>] [line linear|step|quadratic|error width <value>]
      [<plotname>] [line linear|step|quadratic dash yes|no]
      [<plotname>] [line error 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 -plotclear # clear all datasets
$ds9 -plotload foo.dat xy # load new dataset with dimension xy
$ds9 -plotsave bar.dat # save current dataset
$ds9 -plotloadconfig foo.plt # load plot configuration
$ds9 -plotsaveconfig bar.plt # save current plot configuration
$ds9 -plotprint
$ds9 -plotprint destination file
$ds9 -plotprint command "lp"
$ds9 -plotprint filename "foo.ps"
$ds9 -plotprint palette gray
$ds9 -plotpage orientation portrait
$ds9 -plotpage pagescale scaled
$ds9 -plotpage pagesize letter
$ds9 -plotgraph grid yes
$ds9 -plotgraph scale loglog
$ds9 -plotgraph range x auto yes
$ds9 -plotgraph range x min 0
$ds9 -plotgraph range x max 100
$ds9 -plotgraph range y auto yes
$ds9 -plotgraph range y min 0
$ds9 -plotgraph range y max 100
$ds9 -plotgraph labels title {The Title}
$ds9 -plotgraph labels xaxis {X}
$ds9 -plotgraph labels yaxis {Y}
$ds9 -plotfont numbers font times
$ds9 -plotfont numbers size 12
$ds9 -plotfont numbers weight bold
$ds9 -plot font numbers slant italic
$ds9 -plotfont labels font times
$ds9 -plotfont title font times
# edit current dataset
$ds9 -plotdataset 2 # set current dataset to the second dataset
loaded
$ds9 -plotview discrete yes
$ds9 -plotcolor discrete red
$ds9 -plotline discrete cross
$ds9 -plotline step width 2
$ds9 -plotline step dash yes
$ds9 -plot line error 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
```

pow

Select Power scale for current frame.

```
Syntax:
-pow

Example:
$ds9 -pow
```

prefs

Controls various preference settings.

```
Syntax:
prefs [clear]
      [bgcolor <color>]
      [nancolor <color>]

Example:
$ds9 -prefs clear
$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
```

psprint

For MacOSX and Windows, invokes postscript printing. For all others, same as print. Please see print for further details.

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]
       [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 resolution 75
```

private

use private colormap, valid for pseudocolor 8 mode.

```
Syntax:
-private

Example:
$ds9 -private
```

raise

Raise in the window stacking order.

```
Syntax:
-raise

Example:
$ds9 -raise
```

regions

Controls regions in the current frame.

```
Syntax:
-regions [<filename>]
         [load [all] <filename>]
         [save <filename>]
         [list [close]]
         [show yes|no]
         [showtext yes|no]
         [centroid]
         [centroid auto yes|no]
         [centroid radius <value>|iteration <value>]
         [getinfo]
         [move front]
         [move back]
         [select all]
         [select none]
         [select group <groupname>]
         [delete all]
         [delete select]
         [format ds9|xml|ciao|saotng|saoimage|pros|xy]
         [system image|physical|wcs|wcsa...wcsz]
         [sky fk4|fk5|icrs|galactic|ecliptic]
         [skyformat degrees|sexagesimal]
         [strip yes|no]
         [shape <shape>]
         [color <color>
         [width <width>]
         [delim [nl|<char>]]
         [command <marker command>]
         [composite]
         [dissolve]
         [template <filename>]
         [template <filename> at <ra> <dec> <coordsys> <skyframe>]
         [savetemplate <filename>]
```

```
          [group <tag>]
          [group <tag> color <color>]
          [group <tag> copy]
          [group <tag> delete]
          [group <tag> cut]
          [group <tag> font <font>]
          [group <tag> move <int> <int>]
          [group <tag> movefront]
          [group <tag> moveback]
          [group <tag> property <property> yes|no]
          [group <tag> select]
          [copy]
          [cut]
          [paste image|physical|wcs|wcsa...wcsz]
          [undo]
```

```
Example:
$ds9 -regions foo.reg
$ds9 -regions -format ciao bar.reg # load as ciao format
$ds9 -regions foo.fits # FITS regions files do not need a format
specification
$ds9 -regions load foo.reg # load foo.reg into current frame
$ds9 -regions load all foo.reg # load foo.reg into all frames
$ds9 -regions load ’*.reg’# expand *.reg and load into current frame
$ds9 -regions load all ’*.reg’ # expand *.reg and load into all
frames
$ds9 -regions save foo.reg
$ds9 -regions list
$ds9 -regions list close
$ds9 -regions show yes
$ds9 -regions showtext no
$ds9 -regions centroid
$ds9 -regions centroid auto yes
$ds9 -regions centroid radius 10
$ds9 -regions centroid iteration 20
$ds9 -regions getinfo
$ds9 -regions move back
$ds9 -regions move front
$ds9 -regions select all
$ds9 -regions select none
$ds9 -regions select group foo
$ds9 -regions delete all
$ds9 -regions delete select
$ds9 -regions format ds9
$ds9 -regions system wcs
$ds9 -regions sky fk5
$ds9 -regions skyformat degrees
```

```
$ds9 -regions delim nl
$ds9 -regions strip yes
$ds9 -regions shape ellipse
$ds9 -regions color red
$ds9 -regions width 3
$ds9 -regions command "circle 100 100 20"
$ds9 -regions composite
$ds9 -regions dissolve
$ds9 -regions template foo.tpl
$ds9 -regions template foo.tpl at 13:29:55.92 +47:12:48.02 fk5
$ds9 -regions savetemplate foo.tpl
$ds9 -regions group foo color red
$ds9 -regions group foo copy
$ds9 -regions group foo delete
$ds9 -regions group foo cut
$ds9 -regions group foo font {times 14 bold}
$ds9 -regions group foo move 100 100
$ds9 -regions group foo movefront
$ds9 -regions group foo moveback
$ds9 -regions group foo property delete no
$ds9 -regions group foo select
$ds9 -regions copy
$ds9 -regions cut
$ds9 -regions paste wcs
$ds9 -regions undo
```

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 []
     [red|green|blue]
     [channel [red|green|blue]]
     [view [red|green|blue] [yes|no]]
     [system <coordsys>]
     [lock scale|bin|colorbar|slice|smooth [yes|no]
     [open|close]
```

```
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
$ds9 -rgb lock slice yes # lock rgb slice channels
$ds9 -rgb lock smooth yes # lock rgb smooth channels
$ds9 -rgb open
$ds9 -rgb close
```

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>]
        [open|close]

Example:
$ds9 -rotate 45
$ds9 -rotate to 30
$ds9 -rotate open
$ds9 -rotate close
```

samp

Enable/Disable SAMP protocol.

```
Syntax:
-samp [yes|no]
      [broadcast [image|table]]
      [send [image|table] <application>]

Example:
$ds9 -samp yes
$ds9 -samp broadcast image
$ds9 -samp send image aladin
```

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 <filename>
-saveimage jpeg [1-100] <filename>
```

```
-saveimage tiff [none|jpeg|packbits|deflate] <filename>
-saveimage png <filename>
-saveimage ppm <filename>
-saveimage mpeg [1-31] <filename>

Example:
$ds9 -saveimage fits ds9.fits
$ds9 -saveimage jpeg 75 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|pow|sqrt|squared|histequ]
       [log exp <value>]
       [datasec yes|no]
       [limits <minvalue> <maxvalue>]
       [mode minmax|<value>|zscale|zmax]
       [scope local|global]
       [open|close]

Example:
$ds9 -scale linear
$ds9 -scale log exp 100
$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
$ds9 -scale open
$ds9 -scale close
```

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
```

skyview

Support for SkyView image server at HEASARC.

```
Syntax:
-skyview [<object>]
         [name <object>]
         [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
         [size <width> <height> degrees|arcmin|arcsec]
         [save yes|no]
         [frame new|current]
         [update frame|crosshair]
         [survey sdssi|sdssr|sdssg|sdssu|sdssg]
         [open|close]

Example:
$ds9 -skyview m31
$ds9 -skyview name m31
$ds9 -skyview coord 00:42:44.404 +41:16:08.78 sexagesimal
$ds9 -skyview size 60 60 arcmin
$ds9 -skyview save yes
$ds9 -skyview frame current
$ds9 -skyview update frame
$ds9 -skyview survey sdssi
$ds9 -skyview open
$ds9 -skyview close
```

sleep

Delays execution for specified number of seconds. Default is 1 second.

```
Syntax:
-sleep [#]
```

```
Example:
$ds9 -sleep
$ds9 -sleep 2
```

smooth

Smooth current image or set smooth parameters.

```
Syntax:
-smooth []
        [yes|no]
        [function boxcar|tophat|gaussian]
        [radius <int>]
        [open|close]

Example:
$ds9 -smooth
$ds9 -smooth yes
$ds9 -smooth function tophat
$ds9 -smooth radius 4
$ds9 -smooth open
$ds9 -smooth close
```

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

Source TCL code from a file.

```
Syntax:
-source filename

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

tcl

Enable tcl commands to be executed via XPA or SAMP. This can be a major security risk and is disabled by default. Please use with caution.

```
Syntax:
-tcl [yes|no]

Example:
$ds9 -tcl yes
```

tile

Controls the tile display mode.

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

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:
$ds9 -update
$ds9 -update 1 100 100 300 400
$ds9 -update now
$ds9 -update now 1 100 100 300 400
```

```
$ds9 -update off # delay refresh of the screen while loading files
$ds9 -update on # be sure to turn it on when you are finished
loading
```

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 and exits.

```
Syntax:
-version

Example:
$ds9 -version
```

view

Controls the GUI and visible RGB frame color channels.

```
Syntax:
-view [layout horizontal|vertical]
      [info yes|no]
      [panner yes|no]
      [magnifier yes|no]
      [buttons yes|no]
      [colorbar yes|no]
      [graph horizontal|vertical yes|no]
      [filename yes|no]
      [object yes|no]
      [minmax yes|no]
      [lowhigh yes|no]
      [frame yes|no]
      [image|physical|wcs|wcsa...wcsz yes|no]
      [red yes|no]
      [green yes|no]
      [blue yes|no]

Example:
$ds9 -view layout vertical
```

```
$ds9 -view info yes
$ds9 -view panner yes
$ds9 -view magnifier yes
$ds9 -view buttons yes
$ds9 -view colorbar yes
$ds9 -view graph horizontal yes
$ds9 -view filename yes
$ds9 -view object yes
$ds9 -view minmax yes
$ds9 -view lowhigh yes
$ds9 -view frame yes
$ds9 -view wcsa yes
$ds9 -view red yes
$ds9 -view green yes
$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 [method xpa|mime]
    [server <url>]
    [internal yes|no]
    [delay #]
    [<url>]
    [connect <url>]
    [disconnect <url>]
    [open|close]

Example:
$ds9 -vo method xpa
$ds9 -vo server "http://foo.bar.edu/list.txt"
$ds9 -vo internal yes
$ds9 -vo delay 15 # keep-alive delay
$ds9 -vo chandra-ed
```

```
$ds9 -vo connect chandra-ed
$ds9 -vo disconnect chandra-ed
$ds9 -vo open
$ds9 -vo close
```

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 [#] <filename>]
     [append [#] <filename>]
     [open|close]

Example:
$ds9 -wcs wcs
$ds9 -wcs system wcs
$ds9 -wcs wcsa
$ds9 -wcs sky fk5
$ds9 -wcs skyformat sexagesimal
$ds9 -wcs align yes
$ds9 -wcs reset
$ds9 -wcs reset 3
$ds9 -wcs replace foo.wcs
$ds9 -wcs replace 3 foo.wcs
$ds9 -wcs append foo.wcs
$ds9 -wcs append 3 foo.wcs
$ds9 -wcs open
$ds9 -wcs close
```

web

Display specified URL in the web display.

```
Syntax:
-web [new|<webname>] [<url>]
     [<webname>] [click back|forward|stop|reload|#]
     [<webname>] [clear]
     [<webname>] [close]
```

```
Example:
$ds9 -web www.cnn.com
$ds9 -web new www.cnn.com
$ds9 -web hvweb www.apple.com
$ds9 -web click back
$ds9 -web click 2
$ds9 -web clear
$ds9 -web close
```

width

Set the width of the image display window. Use the geometry command to set the overall width and height of the ds9 window.

```
Syntax:
-width [<value>]

Example:
$ds9 -width 512
```

xpa

Configure XPA at startup.

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

Example:
$ds9 -xpa no
$ds9 -xpa local
$ds9 -xpa noxpans
```

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 []
        [contrast]
        [sample]
        [line]

Example:
$ds9 -zscale
$ds9 -zscale contrast .25
$ds9 -zscale sample 600
$ds9 -zscale line 120
```

zoom

Controls the current zoom value for the current frame.

```
Syntax:
-zoom [<value>]
      [<value> <value>]
      [to <value>]
      [to <value> <value>]
      [to fit]
      [open|close]

Example:
$ds9 -zoom 2
$ds9 -zoom 2 4
$ds9 -zoom to 4
$ds9 -zoom to 2 4
$ds9 -zoom to fit
$ds9 -zoom open
$ds9 -zoom close
```

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). In general, select the lowest resolution possible, as postscript file size grows by the square of the increase.

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.

`2mass`
`about`
`analysis`
`array`
`bin`
`blink`
`catalog`
`cd`
`cmap`
`colorbar`
`console`
`contour`
`crosshair`
`cursor`
`data`
`datacube`
`dsssao`
`dsseso`
`dssstsci`
`exit`
`file`
`first`
`fits`
`frame`
`grid`
`header`
`height`
`iconify`
`iis`
`imexam`
`lock`
`lower`
`magnifier`
`mask`

match
minmax
mode
nameserver
nvss
orient
pagesetup
pan
pixeltable
plot
prefs
preserve
psprint
print
quit
raise
regions
rgb
rotate
saveimage
savefits
savempeg
scale
shm
single
skyview
sleep
smooth
source
tcl
tile
update
version
view
vo
wcs
web
width
zscale
zoom

2mass

Support for 2MASS Digital Sky Survey.

```
Syntax:
2mass [<object>]
```

```
      [name <object>]
      [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
      [size <width> <height> degrees|arcmin|arcsec]
      [save yes|no]
      [frame new|current]
      [update frame|crosshair]
      [survey j|h|k]
      [open|close]

Example:
$xpaget ds9 2mass name
$xpaget ds9 2mass coord
$xpaget ds9 2mass size
$xpaget ds9 2mass save
$xpaget ds9 2mass frame
$xpaget ds9 2mass survey
$xpaset -p ds9 2mass m31
$xpaset -p ds9 2mass name m31
$xpaset -p ds9 2mass coord 00:42:44.404 +41:16:08.78 sexagesimal
$xpaset -p ds9 2mass size 60 60 arcmin
$xpaset -p ds9 2mass save yes
$xpaset -p ds9 2mass frame current
$xpaset -p ds9 2mass update frame
$xpaset -p ds9 2mass survey j
$xpaset -p ds9 2mass open
$xpaset -p ds9 2mass close
```

about

Get DS9 credits.

```
Syntax:
about

Example:
$xpaget ds9 about
```

analysis

Control external analysis tasks. Tasks are numbered as they are loaded, starting with 0. Can also be used to display a message and display text in the text dialog window.

```
Syntax:
analysis [<task number>]
         [<filename>]
         [task <task number>|<task name>]
         [load <filename>]
         [clear]
         [clear][load <filename>]
         [message ok|okcancel|yesno <message>]
         [entry <message>]
         [text]

Example:
$xpaget ds9 analysis
$xpaget ds9 analysis task
$xpaget ds9 analysis entry ’Please enter something’
$xpaget ds9 analysis entry okcancel ’Please enter something’
$xpaset -p ds9 analysis 0 # invoke first analysis task
$xpaset -p ds9 analysis task 0
$xpaset -p ds9 analysis task foobar
$xpaset -p ds9 analysis "{foo bar}"
$xpaset -p ds9 analysis my.ans
$xpaset -p ds9 analysis load my.ans
$xpaset -p ds9 analysis clear
$xpaset -p ds9 analysis clear load my.ans
$xpaset -p ds9 analysis message ok {This is a message}
$xpaset -p ds9 analysis text {this is text}
$cat my.ans | 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 [bigendian|littleendian]
array
[new|mask][[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 # default bigendian
$xpaget ds9 array littleendian
$cat foo.arr | xpaset ds9 array [dim=512,bitpix=16]
$cat foo.arr | xpaset ds9 array mask [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] #
256x256 float with 4 byte head
$cat bar.arr | xpaset ds9 array
```

```
[dim=512,bitpix=32,arch=littleendian] # 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>]
    [about center]
    [buffersize <value>]
    [cols <x> <y>]
    [colsz <x> <y> <z>]
    [factor <value> [<vector>]]
    [depth <value>]
    [filter <string>]
    [function average|sum]
    [to fit]
    [open|close]

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 about center
$xpaset -p ds9 bin buffersize 512
$xpaset -p ds9 bin cols detx dety
$xpaset -p ds9 bin colsz detx dety time
$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 filter ’’
$xpaset -p ds9 bin function sum
$xpaset -p ds9 bin to fit
$xpaset -p ds9 bin open
$xpaset -p ds9 bin close
```

blink

Blink mode parameters. Interval is in seconds.

```
Syntax:
blink []
      [yes|no]
      [interval <value>]

Example:
$xpaget ds9 blink
$xpaget ds9 blink interval
$xpaset -p ds9 blink
$xpaset -p ds9 blink yes
$xpaset -p ds9 blink interval 1
```

catalog
cat

Support for catalogs. The first three commands will create a new catalog search. All other commands operated on the last search created, unless indicated otherwise.

```
Syntax:
catalog []
        [ned|simbad|denis|skybot]

[ascss|cmc|gsc1|gsc2|gsc3|ac|nomad|ppmx|sao|sdss5|sdss6|tycho|ua2|ub1|ucac2]
        [2mass|iras]
        [csc|xmm|rosat]
        [first|nvss]
        [chandralog|cfhtlog|esolog|stlog|xmmlog]
        [cds <catalogname>]
        [cds <catalogid>]
        [load <filename>]
        [load [xml|sb|tsv] <filename>]
        [<catname>] [samp]
        [<catname>] [samp broadcast]
        [<catname>] [samp send <application>]
        [<catname>] [name <object>]
        [<catname>] [coordinate <ra> <dec> <coordsys>]
        [<catname>] [size <width> <height> degrees|arcmin|arcsec]
        [<catname>] [save <filename>]
        [<catname>] [save [xml|sb|tsv] <filename>]
        [<catname>] [header]
        [<catname>] [filter <string>]
        [<catname>] [filter load <filename>]
        [<catname>] [clear]
        [<catname>] [retrieve]
        [<catname>] [cancel]
        [<catname>] [print]
        [<catname>] [close]
        [<catname>] [server
cds|sao|cadc|adac|iucaa|bejing|cambridge|ukirt]
        [<catname>] [symbol [#]
condition|shape|color|text|font|fontsize|fontweight|fontslant <value>]
        [<catname>] [symbol [#] text|size|size2|units|angle <value>]
        [<catname>] [symbol shape {circle point}|{box
point}|{diamond point}|
                    {cross point}|{x point}|{arrow point}|{boxcircle
point}|
                    circle|ellipse|box|text]
        [<catname>] [symbol add| [#] remove]
        [<catname>] [symbol save|load <filename>]
        [<catname>] [sort <columnname> incr|decr]
        [<catname>] [maxrows <number>]
        [<catname>] [allrows]
        [<catname>] [allcols]
        [<catname>] [show|hide]
        [<catname>] [ra <columnname>]
        [<catname>] [dec <columnname>]
        [<catname>] [system <coordsys>]
        [<catname>] [sky <skyframe>]
        [<catname>] [show|hide]
        [<catname>] [edit yes|no]
        [<catname>] [panto yes|no]

Example:
$xpaget ds9 catalog
$xpaget ds9 catalog header
$xpaget ds9 catalog cat2mass header
$xpaset -p ds9 catalog
$xpaset -p ds9 catalog 2mass
$xpaset -p ds9 catalog cds 2mass
$xpaset -p ds9 catalog cds "I/252"
$xpaset -p ds9 catalog load foo.cat
$xpaset -p ds9 catalog load xml foo.xml
$xpaset -p ds9 catalog samp broadcast
$xpaset -p ds9 catalog samp send aladin
$xpaset -p ds9 catalog cat2mass symbol color red
$xpaset -p ds9 catalog name m51
$xpaset -p ds9 catalog coordinate 202.48 47.21 fk5
$xpaset -p ds9 catalog size 22 22 arcmin
$xpaset -p ds9 catalog save bar.cat
$xpaset -p ds9 catalog save xml bar.xml
```

```
$xpaset -p ds9 catalog filter "\$Jmag>10"
$xpaset -p ds9 catalog filter load foo.flt
$xpaset -p ds9 catalog clear
$xpaset -p ds9 catalog retrieve
$xpaset -p ds9 catalog cancel
$xpaset -p ds9 catalog print
$xpaset -p ds9 catalog close
$xpaset -p ds9 catalog server sao
$xpaset -p ds9 catalog symbol condition "\$Jmag>15"
$xpaset -p ds9 catalog symbol 2 shape "boxcircle point"
$xpaset -p ds9 catalog symbol color red
$xpaset -p ds9 catalog font times
$xpaset -p ds9 catalog fontsize 14
$xpaset -p ds9 catalog fontweight bold
$xpaset -p ds9 catalog fontslant italic
$xpaset -p ds9 catalog symbol add
$xpaset -p ds9 catalog symbol 2 remove
$xpaset -p ds9 catalog symbol load foo.sym
$xpaset -p ds9 catalog symbol save bar.sym
$xpaset -p ds9 catalog sort "Jmag" incr
$xpaset -p ds9 catalog maxrows 2000
$xpaset -p ds9 catalog allrows
$xpaset -p ds9 catalog allcols
$xpaset -p ds9 catalog ra RA
$xpaset -p ds9 catalog dec DEC
$xpaset -p ds9 catalog system wcs
$xpaset -p ds9 catalog sky fk5
$xpaset -p ds9 catalog show
$xpaset -p ds9 catalog hide
$xpaset -p ds9 catalog edit yes
$xpaset -p ds9 catalog panto no
```

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 <constrast> <bias>]
     [open|close]

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
$xpaset -p ds9 cmap open
$xpaset -p ds9 cmap close
```

colorbar

Controls colorbar parameters.

```
Syntax:
colorbar [yes|no]
         [horizontal|vertical]
         [orientation horizontal|vertical]
         [numerics yes|no]
         [space value|distance]
         [font times|helvetica|courier]
         [fontsize <value>]
         [fontweight normal|bold]
         [fontslant roman|italic]
         [size]
         [ticks]

Example:
$xpaget ds9 colorbar
$xpaget ds9 colorbar orientation
$xpaget ds9 colorbar numerics
$xpaget ds9 colorbar space
$xpaget ds9 colorbar font
$xpaget ds9 colorbar fontsize
$xpaget ds9 colorbar fontweight
```

```
$xpaget ds9 colorbar fontslant
$xpaget ds9 colorbar size
$xpaget ds9 colorbar ticks
$xpaset -p ds9 colorbar yes
$xpaset -p ds9 colorbar vertical
$xpaset -p ds9 colorbar orientation vertical
$xpaset -p ds9 colorbar numerics yes
$xpaset -p ds9 colorbar space value
$xpaset -p ds9 colorbar font times
$xpaset -p ds9 colorbar fontsize 14
$xpaset -p ds9 colorbar fontweight bold
$xpaset -p ds9 colorbar fontslant italic
$xpaset -p ds9 colorbar size 20
$xpaset -p ds9 colorbar ticks 11
```

console

Display tcl console window.

```
Syntax:
-console

Example:
$xpaset -p ds9 console
```

contour

Controls contours in the current frame.

```
Syntax:
contour []
        [yes|no]
        [<coordsys> [<skyframe>]]
        [clear]
        [generate]
        [load <filename> <coordsys> <skyframe> <color> <width>
yes|no]
        [save <filename> <coordsys> <skyframe>]
        [convert]
        [loadlevels <filename>]
        [savelevels <filename>]
        [copy]
        [paste <coordsys> <color> <width> yes|no]
        [color <color>]
        [width <width>]
        [dash yes|no]
        [smooth <smooth>]
        [method block|smooth]
        [nlevels <number of levels>]
        [scale linear|log|pow|squared|sqrt|histequ]
        [log exp <value>]
        [mode minmax|<value>|zscale|zmax]
        [limits <min> <max>]
        [levels <value value value...>]
        [open|close]

Example:
$xpaget ds9 contour
$xpaget ds9 contour wcs fk5
$xpaget ds9 contour color
$xpaget ds9 contour width
$xpaget ds9 contour dash
$xpaget ds9 contour smooth
$xpaget ds9 contour method
$xpaget ds9 contour nlevels
$xpaget ds9 contour scale
$xpaget ds9 contour log exp
$xpaget ds9 contour mode
$xpaget ds9 contour limits
$xpaget ds9 contour levels
$xpaset -p ds9 contour
$xpaset -p ds9 contour yes
$xpaset -p ds9 contour clear
$xpaset -p ds9 contour generate
$xpaset -p ds9 contour load ds9.con wcs fk5 yellow 2 no # solid line
$xpaset -p ds9 contour load ds9.con wcs fk5 red 2 yes # dashed line
$xpaset -p ds9 contour save ds9.con wcs fk5
$xpaset -p ds9 contour convert
$xpaset -p ds9 contour loadlevels ds9.lev
$xpaset -p ds9 contour savelevels ds9.lev
$xpaset -p ds9 contour copy
$xpaset -p ds9 contour paste wcs red 2 no
$xpaset -p ds9 contour color yellow
$xpaset -p ds9 contour width 2
$xpaset -p ds9 contour dash yes
$xpaset -p ds9 contour smooth 5
$xpaset -p ds9 contour method smooth
$xpaset -p ds9 contour nlevels 10
$xpaset -p ds9 contour scale sqrt
$xpaset -p ds9 contour log exp 1000
$xpaset -p ds9 contour mode zscale
$xpaset -p ds9 contour limits 1 100
$xpaset -p ds9 contour levels "{1 10 100 1000}"
$xpaset -p ds9 contour open
$xpaset -p ds9 contour close
```

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 <coordsys> [<skyframe>][<skyformat>]]

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 of yes indicates to strip the coordinates from the output and just list the data values. The default is yes.

```
Syntax:
data [<coordsys> [<skyframe>] <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 #]
         [axis #]
         [open|close]

Example:
$xpaget ds9 datacube
$xpaget ds9 datacube interval
$xpaget ds9 datacube axis
$xpaset -p ds9 datacube play
$xpaset -p ds9 datacube last
$xpaset -p ds9 datacube 3
$xpaset -p ds9 datacube interval 2
$xpaset -p ds9 datacube axis 3
$xpaset -p ds9 datacube open
$xpaset -p ds9 datacube close
```

dsssao

Support for Digital Sky Survey at SAO.

```
Syntax:
dsssao [<object>]
       [name <object>]
       [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
       [size <width> <height> degrees|arcmin|arcsec]
       [save yes|no]
       [frame new|current]
       [update frame|crosshair]
       [open|close]

Example:
$xpaget ds9 dsssao name
$xpaget ds9 dsssao coord
$xpaget ds9 dsssao size
$xpaget ds9 dsssao save
$xpaget ds9 dsssao frame
$xpaset -p ds9 dsssao m31
$xpaset -p ds9 dsssao name m31
$xpaset -p ds9 dsssao coord 00:42:44.404 +41:16:08.78 sexagesimal
$xpaset -p ds9 dsssao size 60 60 arcmin
```

```
$xpaset -p ds9 dsssao save yes
$xpaset -p ds9 dsssao frame current
$xpaset -p ds9 dsssao update frame
$xpaset -p ds9 dsssao open
$xpaset -p ds9 dsssao close
```

dsseso

Support for Digital Sky Survey at ESO.

```
Syntax:
dsseso [<object>]
       [name <object>]
       [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
       [size <width> <height> degrees|arcmin|arcsec]
       [save yes|no]
       [frame new|current]
       [update frame|crosshair]
       [survey DSS1|DSS2-red|DSS2-blue|DSS2-infrared]
       [open|close]

Example:
$xpaget ds9 dsseso name
$xpaget ds9 dsseso coord
$xpaget ds9 dsseso size
$xpaget ds9 dsseso save
$xpaget ds9 dsseso frame
$xpaget ds9 dsseso survey
$xpaset -p ds9 dsseso m31
$xpaset -p ds9 dsseso name m31
$xpaset -p ds9 dsseso coord 00:42:44.404 +41:16:08.78 sexagesimal
$xpaset -p ds9 dsseso size 60 60 arcmin
$xpaset -p ds9 dsseso save yes
$xpaset -p ds9 dsseso frame current
$xpaset -p ds9 dsseso update frame
$xpaset -p ds9 dsseso survey DSS2-red
$xpaset -p ds9 dsseso open
$xpaset -p ds9 dsseso close
```

dssstsci

Support for Digital Sky Survey at STSCI.

```
Syntax:
dssstsci [<object>]
         [name <object>]
         [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
         [size <width> <height> degrees|arcmin|arcsec]
```

```
          [save yes|no]
          [frame new|current]
          [update frame|crosshair]
          [survey poss2ukstu_red|poss2ukstu_ir|poss2ukstu_blue]
          [survey poss1_blue|poss1_red]
          [survey all|quickv|phase2_gsc2|phase2_gsc1]
          [open|close]

Example:
$xpaget ds9 dssstsci name
$xpaget ds9 dssstsci coord
$xpaget ds9 dssstsci size
$xpaget ds9 dssstsci save
$xpaget ds9 dssstsci frame
$xpaget ds9 dssstsci survey
$xpaset -p ds9 dssstsci m31
$xpaset -p ds9 dssstsci name m31
$xpaset -p ds9 dssstsci coord 00:42:44.404 +41:16:08.78 sexagesimal
$xpaset -p ds9 dssstsci size 60 60 arcmin
$xpaset -p ds9 dssstsci save yes
$xpaset -p ds9 dssstsci frame current
$xpaset -p ds9 dssstsci update frame
$xpaset -p ds9 dssstsci survey all
$xpaset -p ds9 dssstsci open
$xpaset -p ds9 dssstsci close
```

exit
quit

Quits DS9.

```
Syntax:
exit
quit

Example:
$xpaset -p ds9 exit
```

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|mask][<filename>]
     [new|mask][fits <filename>]
     [new|mask][sfits <filename> <filename>]
     [new|mask][medatacube <filename>]
     [new|mask][mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] <filename>]
     [new|mask][mosaic [iraf|wcs|wcsa...wcsz] <filename>]
     [new|mask][smosaic [iraf|wcs|wcsa...wcsz] <filename>
<filename>]
     [new][multiframe <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|mask][array
<file>[[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:
$xpaget ds9 file
$xpaset -p ds9 file foo.fits
$xpaset -p ds9 file mask 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 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 multiframe foo.fits
$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
```

first

Support for VLA First Sky Survey.

```
Syntax:
first [<object>]
      [name <object>]
      [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
      [size <width> <height> degrees|arcmin|arcsec]
      [save yes|no]
      [frame new|current]
      [update frame|crosshair]
      [open|close]

Example:
$xpaget ds9 first name
$xpaget ds9 first coord
$xpaget ds9 first size
$xpaget ds9 first save
$xpaget ds9 first frame
$xpaset -p ds9 first m31
$xpaset -p ds9 first name m31
$xpaset -p ds9 first coord 00:42:44.404 +41:16:08.78 sexagesimal
$xpaset -p ds9 first size 60 60 arcmin
$xpaset -p ds9 first save yes
$xpaset -p ds9 first frame current
$xpaset -p ds9 first update frame
$xpaset -p ds9 first open
$xpaset -p ds9 first close
```

fits

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

```
Syntax:
fits [size|width|height|depth|bitpix]
     [size [wcs|wcsa...wcsz] [fk4|fk5|icrs|galactic|ecliptic]
[degrees|arcmin|arcsecs]]
     [type]
     [header [#] [keyword <string>]]
     [image|table|resample] [gz]
     [new|mask][<options>]
     [new|mask][medatacube <options>]
     [new|mask][mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] <options>]
     [new|mask][mosaic [iraf|wcs|wcsa...wcsz] <options>]
     [new][rgbcube <options>]
     [new][rgbimage <options>]
     [save resample gz <filename>]

Example:
$xpaget ds9 fits > foo.fits
$xpaget ds9 fits size
$xpaget ds9 fits width
$xpaget ds9 fits height
$xpaget ds9 fits depth
$xpaget ds9 fits bitpix
$xpaget ds9 fits size wcs fk5 arcmin
$xpaget ds9 fits type
$xpaget ds9 fits header # primary
$xpaget ds9 fits header 2 # hdu 2
$xpaget ds9 fits header -2 # hdu 2 with inherit
$xpaget ds9 fits header keyword "'BITPIX'"
$xpaget ds9 fits header 1 keyword "'BITPIX'"
$xpaget ds9 fits image > foo.fits
$xpaget ds9 fits image gz > foo.fits.gz
$xpaget ds9 fits table > bar.fits
$xpaget ds9 fits table gz > bar.fits.gz
$xpaget ds9 fits resample > bar.fits
$xpaget 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 mosaic iraf
$cat bar.fits | xpaset ds9 fits mosaic wcs
$cat rgb.fits | xpaset ds9 fits rgbcube
$cat rgb.fits | xpaset 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]]
      [move first]
      [move back]
      [move forward]
      [move last]
      [first]
      [prev]
      [next]
      [last]
      [frameno #]
      [#]
      [has
[amplifier|datamin|datasec|detector|grid|iis|irafmin|physical|smooth]]
      [has contour [aux]]]
      [has fits [ |bin|cube|mosaic]]
      [has marker [highlite|paste|select|undo]]
      [has system <coordsys>]
      [has wcs [<wcssys>|equatorial <wcssys>|linear <wcssys>]]

Example:
$xpaget ds9 frame # returns the id of the current frame
$xpaget ds9 frame frameno # returns the id of the current frame
$xpaget ds9 frame all # returns the id of all frames
$xpaget ds9 frame active # returns the id of all active frames
$xpaget ds9 frame has amplifier
$xpaget ds9 frame has datamin
$xpaget ds9 frame has datasec
$xpaget ds9 frame has detector
$xpaget ds9 frame has grid
$xpaget ds9 frame has iis
$xpaget ds9 frame has irafmin
$xpaget ds9 frame has physical
$xpaget ds9 frame has smooth
$xpaget ds9 frame has contour
$xpaget ds9 frame has contour aux
$xpaget ds9 frame has fits
$xpaget ds9 frame has fits bin
$xpaget ds9 frame has fits cube
$xpaget ds9 frame has fits mosaic
$xpaget ds9 frame has marker highlite
$xpaget ds9 frame has marker paste
$xpaget ds9 frame has marker select
```

```
$xpaget ds9 frame has marker undo
$xpaget ds9 frame has system physical
$xpaget ds9 frame has wcs wcsa
$xpaget ds9 frame has wcs equatorial wcsa
$xpaget ds9 frame has wcs linear wcsa
$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 move first # move frame to first in order
$xpaset -p ds9 frame move back # move frame back in order
$xpaset -p ds9 frame move forward # move frame forward in order
$xpaset -p ds9 frame move last # move frame to last in order
$xpaset -p ds9 frame first # goto first frame
$xpaset -p ds9 frame prev # goto prev frame
$xpaset -p ds9 frame next # goto next 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. For grid numeric format syntax, click here.

```
Syntax:
grid  []
      [yes|no]
      [type analysis|publication]
      [system <coordsys>]
      [sky <skyframe>]
      [skyformat <skyformat>]
      [grid yes|no]
      [grid color <color>]
      [grid width <value>]
      [grid style 0|1]
      [grid gap1 <value>]
      [grid gap2 <value>]
      [axes yes|no]
      [axes color <color>]
```

```
[axes width <value>]
[axes style 0|1]
[axes type interior|exterior]
[format1 <format>]
[format2 <format>]
[tick yes|no]
[tick color <color>]
[tick width <value>]
[tick style 0|1]
[border yes|no]
[border color <color>]
[border width <value>]
[border style 0|1]
[numlab yes|no]
[numlab font times|helvetica|courier]
[numlab fontsize <value>]
[numlab fontweight normal|bold]
[numlab fontslant roman|italic]
[numlab color <color>]
[numlab gap1 <value>]
[numlab gap2 <value>]
[numlab type interior|exterior]
[numlab vertical yes|no]
[title yes|no]
[title text <text>]
[title def yes|no]
[title gap <value>]
[title font times|helvetica|courier]
[title fontsize <value>]
[title fontweight normal|bold]
[title fontslant roman|italic]
[title color <color>]
[textlab yes|no]
[textlab text1 <text>]
[textlab def1 yes|no]
[textlab gap1 <value>]
[textlab text2 <text>]
[textlab def2 yes|no]
[textlab gap2 <value>]
[textlab font times|helvetica|courier]
[textlab fontsize <value>]
[textlab fontweight normal|bold]
[textlab fontslant roman|italic]
[textlab color <color>]
[reset]
[load <filename>]
[save <filename>]
```

```
      [open|close]

Example:
$xpaget ds9 grid
$xpaget ds9 grid type
$xpaget ds9 grid system
$xpaget ds9 grid sky
$xpaget ds9 grid skyformat
$xpaget ds9 grid grid
$xpaget ds9 grid grid color
$xpaget ds9 grid grid width
$xpaget ds9 grid grid style
$xpaget ds9 grid grid gap1
$xpaget ds9 grid grid gap2
$xpaget ds9 grid axes
$xpaget ds9 grid axes color
$xpaget ds9 grid axes width
$xpaget ds9 grid axes style
$xpaget ds9 grid axes type
$xpaget ds9 grid format1
$xpaget ds9 grid format2
$xpaget ds9 grid tick
$xpaget ds9 grid tick color
$xpaget ds9 grid tick width
$xpaget ds9 grid tick style
$xpaget ds9 grid border
$xpaget ds9 grid border color
$xpaget ds9 grid border width
$xpaget ds9 grid border style
$xpaget ds9 grid numlab
$xpaget ds9 grid numlab font
$xpaget ds9 grid numlab fontsize
$xpaget ds9 grid numlab fontweight
$xpaget ds9 grid numlab fontslant
$xpaget ds9 grid numlab color
$xpaget ds9 grid numlab gap1
$xpaget ds9 grid numlab gap2
$xpaget ds9 grid numlab type
$xpaget ds9 grid numlab vertical
$xpaget ds9 grid title
$xpaget ds9 grid title text
$xpaget ds9 grid title def
$xpaget ds9 grid title gap
$xpaget ds9 grid title font
$xpaget ds9 grid title fontsize
$xpaget ds9 grid title fontweight
$xpaget ds9 grid title fontslant
```

```
$xpaget ds9 grid title color
$xpaget ds9 grid textlab
$xpaget ds9 grid textlab text1
$xpaget ds9 grid textlab def1
$xpaget ds9 grid textlab gap1
$xpaget ds9 grid textlab text2
$xpaget ds9 grid textlab def2
$xpaget ds9 grid textlab gap2
$xpaget ds9 grid textlab font
$xpaget ds9 grid textlab fontsize
$xpaget ds9 grid textlab fontweight
$xpaget ds9 grid textlab fontslant
$xpaget ds9 grid textlab color
$xpaset -p ds9 grid
$xpaset -p ds9 grid yes
$xpaset -p ds9 grid type analysis
$xpaset -p ds9 grid system wcs
$xpaset -p ds9 grid sky fk5
$xpaset -p ds9 grid skyformat degrees
$xpaset -p ds9 grid grid yes
$xpaset -p ds9 grid grid color red
$xpaset -p ds9 grid grid width 2
$xpaset -p ds9 grid grid style 1
$xpaset -p ds9 grid grid gap1 10
$xpaset -p ds9 grid grid gap2 10
$xpaset -p ds9 grid axes yes
$xpaset -p ds9 grid axes color red
$xpaset -p ds9 grid axes width 2
$xpaset -p ds9 grid axes style 1
$xpaset -p ds9 grid axes type exterior
$xpaset -p ds9 grid format1 d.2
$xpaset -p ds9 grid format2 d.2
$xpaset -p ds9 grid tick yes
$xpaset -p ds9 grid tick color red
$xpaset -p ds9 grid tick width 2
$xpaset -p ds9 grid tick style 1
$xpaset -p ds9 grid border yes
$xpaset -p ds9 grid border color red
$xpaset -p ds9 grid border width 2
$xpaset -p ds9 grid border style 1
$xpaset -p ds9 grid numlab yes
$xpaset -p ds9 grid numlab font courier
$xpaset -p ds9 grid numlab fontsize 12
$xpaset -p ds9 grid numlab fontweight bold
$xpaset -p ds9 grid numlab fontslant italic
$xpaset -p ds9 grid numlab color red
$xpaset -p ds9 grid numlab gap1 10
```

```
$xpaset -p ds9 grid numlab gap2 10
$xpaset -p ds9 grid numlab type exterior
$xpaset -p ds9 grid numlab vertical yes
$xpaset -p ds9 grid title yes
$xpaset -p ds9 grid title text {Hello World}
$xpaset -p ds9 grid title def yes
$xpaset -p ds9 grid title gap 10
$xpaset -p ds9 grid title font courier
$xpaset -p ds9 grid title fontsize 12
$xpaset -p ds9 grid title fontweight bold
$xpaset -p ds9 grid title fontslant italic
$xpaset -p ds9 grid title color red
$xpaset -p ds9 grid textlab yes
$xpaset -p ds9 grid textlab text1 {Hello World}
$xpaset -p ds9 grid textlab def1 yes
$xpaset -p ds9 grid textlab gap1 10
$xpaset -p ds9 grid textlab text2 {Hello World}
$xpaset -p ds9 grid textlab def2 yes
$xpaset -p ds9 grid textlab gap2 10
$xpaset -p ds9 grid textlab font courier
$xpaset -p ds9 grid textlab fontsize 12
$xpaset -p ds9 grid textlab fontweight boldj
$xpaset -p ds9 grid textlab fontslant italic
$xpaset -p ds9 grid textlab color red
$xpaset -p ds9 grid reset
$xpaset -p ds9 grid load foo.grd
$xpaset -p ds9 grid save foo.grd
$xpaset -p ds9 grid open
$xpaset -p ds9 grid close
```

header

Display current fits header dialog. Optional extension number maybe specified. Please note, this differs from xpa fits header.

```
Syntax:
header [<value>]
       [close [<value>]]

Example:
$xpaset -p ds9 header
$xpaset -p ds9 header 2
$xpaset -p ds9 header close
```

height

Set the height of the image display window.

```
Syntax:
height [<value>]

Example:
$xpaget ds9 height
$xpaset -p ds9 height 512
```

iconify

Toggles iconification.

```
Syntax:
iconify []
        [yes|no]

Example:
$xpaget ds9 iconify
$xpaset -p ds9 iconify
$xpaset -p ds9 iconify yes
```

iis

Set/Get IIS Filename. Optional mosaic number maybe supplied.

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

Example:
$xpaget ds9 iis filename
$xpaget ds9 iis filename 4
$xpaset -p ds9 iis filename foo.fits
$xpaset -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 <coordsys> [<skyframe>] [<skyformat>]]
       [key] [coordinate <coordsys> [<skyframe>] [<skyformat>]]
       [any] [coordinate <coordsys> [<skyframe>] [<skyformat>]]
       [] [data [width][height]]
       [key] [data [width][height]]
       [any] [data [width][height]]
```

```
Example:
$xpaget ds9 imexam coordinate image
$xpaget ds9 imexam key coordinate image # return coordinate and key
event
$xpaget ds9 imexam any coordinate image # return coordinate and
key/mouse event
$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 key data # return data value and key event
$xpaget ds9 imexam any data # return data value and key/mouse event
$xpaget ds9 imexam data 3 3 # return all data in 3x3 box about
selected point
```

lock

Lock frames.

```
Syntax:
lock [crosshair none|wcs|wcsa...wcsz|physical|image]

Example:
$xpaset -p ds9 lock crosshair wcs
```

lower

Lower in the window stacking order.

```
Syntax:
lower

Example:
$xpaset -p ds9 lower
```

pmagnifier

Controls the magnifier settings.

```
Syntax:
magnifier [color <color>]
          [zoom <value>]
          [cursor yes|no]
          [region yes|no]

Example:
$xpaget ds9 magnifier color
$xpaget ds9 magnifier zoom
```

```
$xpaget ds9 magnifier cursor
$xpaget ds9 magnifier region
$xpaset -p ds9 magnifier color yellow
$xpaset -p ds9 magnifier zoom 2
$xpaset -p ds9 magnifier cursor no
$xpaset -p ds9 magnifier region no
```

mask

Controls mask parameters.

```
Syntax:
mask [color <color>]
     [mark 1|0]
     [transparency <value>]
     [clear]
     [open|close]

Example:
$xpaget ds9 mask color
$xpaget ds9 mask mark
$xpaget ds9 mask transparency
$xpaset -p ds9 mask color red
$xpaset -p ds9 mask mark 0
$xpaset -p ds9 mask transparency 50
$xpaset -p ds9 mask clear
$xpaset -p ds9 mask open
$xpaset -p ds9 mask close
```

match

Match all other frames to the current frame.

```
Syntax:
match [frames wcs|physical|image]
      [colorbars]
      [scales]
      [bin]

Example:
$xpaset -p ds9 match frames wcs
$xpaset -p ds9 match colorbars
$xpaset -p ds9 match scales
$xpaset -p ds9 match bin
```

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

Example:
$xpaget ds9 minmax mode
$xpaget ds9 minmax interval
$xpaset -p ds9 minmax scan
$xpaset -p ds9 minmax mode scan
$xpaset -p ds9 minmax interval 10
```

mode

Controls the first mouse button mode.

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

Example:
$xpaget 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]
           [pan]
           [crosshair]
           [open|close]
Example:
$xpaget ds9 nameserver
$xpaget ds9 nameserver server
$xpaget ds9 nameserver skyformat
```

```
$xpaget ds9 nameserver m31
$xpaset -p ds9 nameserver m31
$xpaset -p ds9 nameserver name m31
$xpaset -p ds9 nameserver server ned-sao
$xpaset -p ds9 nameserver skyformat sexagesimal
$xpaset -p ds9 nameserver pan
$xpaset -p ds9 nameserver crosshair
$xpaset -p ds9 nameserver open
$xpaset -p ds9 nameserver close
```

nvss

Support for NRAO VLA Sky Survey.

```
Syntax:
nvss [<object>]
     [name <object>]
     [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
     [size <width> <height> degrees|arcmin|arcsec]
     [save yes|no]
     [frame new|current]
     [update frame|crosshair]
     [open|close]

Example:
$xpaget ds9 nvss name
$xpaget ds9 nvss coord
$xpaget ds9 nvss size
$xpaget ds9 nvss save
$xpaget ds9 nvss frame
$xpaset -p ds9 nvss m31
$xpaset -p ds9 nvss name m31
$xpaset -p ds9 nvss coord 00:42:44.404 +41:16:08.78 sexagesimal
$xpaset -p ds9 nvss size 60 60 arcmin
$xpaset -p ds9 nvss save yes
$xpaset -p ds9 nvss frame current
$xpaset -p ds9 nvss update frame
$xpaset -p ds9 nvss open
$xpaset -p ds9 nvss close
```

orient

Controls the orientation of the current frame.

```
Syntax:
orient [none|x|y|xy]
       [open|close]
```

```
Example:
$xpaget ds9 orient
$xpaset -p ds9 orient xy
$xpaset -p ds9 orient open
$xpaset -p ds9 orient close
```

pagesetup

Controls Page Setup options.

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

Example:
$xpaget ds9 pagesetup orientation
$xpaget ds9 pagesetup pagescale
$xpaget ds9 pagesetup pagesize
$xpaset -p ds9 pagesetup orientation portrait
$xpaset -p ds9 pagesetup pagescale scaled
$xpaset -p ds9 pagesetup pagesize poster
```

pan

Controls the current image cursor location for the current frame.

```
Syntax:
pan [x y <coordsys> [<skyframe>][<skyformat>]]
    [to x y <coordsys> [<skyframe>][<skyformat>]
    [open|close]

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
$xpaset -p ds9 pan open
$xpaset -p ds9 pan close
```

pixeltable

Display/Hide the pixel table.

```
Syntax:
pixeltable []
           [yes|open]
           [no|close]

Example:
$xpaget ds9 pixeltable
$xpaset -p ds9 pixeltable
$xpaset -p ds9 pixeltable yes
$xpaset -p ds9 pixeltable open
$xpaset -p ds9 pixeltable close
```

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>]]
     [new [name <plotname>] <title> <xaxis label> <yaxis label>
 xy|xyex|xyey|xyexey]
# 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>] [print destination 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|courier]
     [<plotname>] [font numbers|labels|title size <value>]
     [<plotname>] [font numbers|labels|title weight normal|bold]
     [<plotname>] [font numbers|labels|title slant roman|italic]
# edit current dataset
plot [<plotname>] [dataset #]
     [<plotname>] [view discrete|linear|step|quadratic|error yes|no]
     [<plotname>] [color discrete|linear|step|quadratic|error
<color>]
     [<plotname>] [line discrete circle|diamond|plus|cross]
     [<plotname>] [line linear|step|quadratic|error width <value>]
     [<plotname>] [line linear|step|quadratic dash yes|no]
     [<plotname>] [line error style 1|2]

Example:
# return all plotnames
$xpaget ds9 plot
# 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 weight bold
$xpaset -p ds9 plot font numbers slant italic
$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 error style 2
```

prefs

Controls various preference settings.

```
Syntax:
prefs [clear]
      [bgcolor <color>]
      [nancolor <color>]
```

```
Example:
$xpaget ds9 prefs bgcolor
$xpaget ds9 prefs nancolor
$xpaget ds9 prefs magnifier
$xpaset -p ds9 prefs clear
$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
```

psprint

For MacOSX and Windows, invokes postscript printing. For all others, same as print. Please see print for further details.

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]
      [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 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 resolution 75
```

raise

Raise in the window stacking order.

```
Syntax:
raise

Example:
$xpaset -p ds9 raise
```

regions

Controls regions in the current frame.

```
Syntax:
regions [<filename>]
        [load [all] <filename>]
        [save <filename>]
        [list [close]]
        [show yes|no]
        [showtext yes|no]
        [centroid]
        [centroid auto yes|no]
        [centroid radius <value>|iteration <value>]
        [getinfo]
        [move front]
        [move back]
        [select all]
        [select none]
        [select group <groupname>]
        [delete all]
        [delete select]
        [format ds9|xml|ciao|saotng|saoimage|pros|xy]
        [system image|physical|wcs|wcsa...wcsz]
        [sky fk4|fk5|icrs|galactic|ecliptic]
        [skyformat degrees|sexagesimal]
        [strip yes|no]
        [shape <shape>]
        [color &ltcolor>]
        [width <width>]
        [delim [nl|<char>]]
        [command <marker command>]
        [composite]
        [dissolve]
        [template <filename>]
        [template <filename> at <ra> <dec> <coordsys> <skyframe>]
        [savetemplate <filename>]
        [groups]
        [group <tag>]
        [group <tag> color &ltcolor>]
        [group <tag> copy]
        [group <tag> delete]
        [group <tag> cut]
        [group <tag> font <font>]
        [group <tag> move <int> <int>]
        [group <tag> movefront]
        [group <tag> moveback]
        [group <tag> property <property> yes|no]
        [group <tag> select]
        [copy]
        [cut]
        [paste image|physical|wcs|wcsa...wcsz]
        [undo]
        [include|exclude|source|background|selected]
        [-format ds9|ciao|saotng|saoimage|pros|xy]
        [-system image|physical|wcs|wcsa...wcsz]
        [-sky fk4|fk5|icrs|galactic|ecliptic]
        [-skyformat degrees|sexagesimal]
        [-delim [nl|<char>]]
        [-prop select|edit|move|rotate|delete|fixed|include|source
1|0]
        [-group <tag>]
        [-strip yes|no]
        [-wcs yes|no]

Example:
$xpaget ds9 regions
$xpaget ds9 regions -format ds9 -system wcs -sky fk5 -skyformat
sexagesimal -prop edit 1 -group foo
$xpaget ds9 regions show
$xpaget ds9 regions showtext
$xpaget ds9 regions centroid auto
$xpaget ds9 regions centroid radius
$xpaget ds9 regions centroid iteration
$xpaget ds9 regions selected
```

```
$xpaget ds9 regions format
$xpaget ds9 regions system
$xpaget ds9 regions sky
$xpaget ds9 regions skyformat
$xpaget ds9 regions strip
$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 groups
$cat foo.reg | xpaset ds9 regions -format xy -system wcs -sky fk5
$cat bar.reg | xpaget ds9 regions -format ds9
$echo "image; circle 100 100 20" | xpaset ds9 regions
$echo "image; circle 100 100 20" | xpaset ds9 regions
$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 foo.reg
$xpaset -p ds9 regions -format ciao bar.reg # load as ciao format
$xpaset -p ds9 regions foo.fits # FITS regions files do not need a
format specification
$xpaset -p ds9 regions load foo.reg # load foo.reg into current
frame
$xpaset -p ds9 regions load all foo.reg # load foo.reg into all
frames
$xpaset -p ds9 regions load ’*.reg’# expand *.reg and load into
current frame
$xpaset -p ds9 regions load all ’*.reg’ # expand *.reg and load into
all frames
$xpaset -p ds9 regions save foo.reg
$xpaset -p ds9 regions list
$xpaset -p ds9 regions list close
$xpaset -p ds9 regions show yes
$xpaset -p ds9 regions showtext no
$xpaset -p ds9 regions centroid
$xpaset -p ds9 regions centroid auto yes
$xpaset -p ds9 regions centroid radius 10
```

```
$xpaset -p ds9 regions centroid iteration 20
$xpaset -p ds9 regions getinfo
$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 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 shape ellipse
$xpaset -p ds9 regions color red
$xpaset -p ds9 regions width 3
$xpaset -p ds9 regions command {circle 100 100 20}
$xpaset -p ds9 regions composite
$xpaset -p ds9 regions dissolve
$xpaset -p ds9 regions template foo.tpl
$xpaset -p ds9 regions template foo.tpl at 13:29:55.92 +47:12:48.02
fk5
$xpaset -p ds9 regions savetemplate foo.tpl
$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
$xpaset -p ds9 regions copy
$xpaset -p ds9 regions cut
$xpaset -p ds9 regions paste wcs
$xpaset -p ds9 regions undo
```

rgb

Create RGB frame and control RGB frame parameters.

```
Syntax:
rgb  []
     [red|green|blue]
     [channel [red|green|blue]]
     [view [red|green|blue] [yes|no]]
     [system <coordsys>]
     [lock scale|bin|colorbar|slice|smooth [yes|no]]
     [open|close]

Example:
$xpaget ds9 rgb channel
$xpaget ds9 rgb lock bin
$xpaget ds9 rgb lock scale
$xpaget ds9 rgb lock colorbar
$xpaget ds9 rgb lock slice
$xpaget ds9 rgb lock smooth
$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
$xpaset -p ds9 rgb lock slice yes # lock rgb slice channels
$xpaset -p ds9 rgb lock smooth yes # lock rgb smooth channels
$xpaset -p ds9 rgb open
$xpaset -p ds9 rgb close
```

rotate

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

```
Syntax:
rotate [<value>]
       [to <value>]
       [open|close]
Example:
$xpaget ds9 rotate
$xpaset -p ds9 rotate 45
$xpaset -p ds9 rotate to 30
$xpaset -p ds9 rotate open
$xpaset -p ds9 rotate close
```

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 <filename>
saveimage jpeg [1-100] <filename>
saveimage tiff [none|jpeg|packbits|deflate] <filename>
saveimage png <filename>
saveimage ppm <filename>
saveimage mpeg [1-31] <filename>

Example:
$xpaset -p ds9 saveimage fits ds9.fits
$xpaset -p ds9 saveimage jpeg 75 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|pow|sqrt|squared|histequ]
      [log exp <value>]
      [datasec yes|no]
      [limits <minvalue> <maxvalue>]
      [mode minmax|<value>|zscale|zmax]
      [scope local|global]
      [open|close]
Example:
$xpaget ds9 scale
$xpaget ds9 scale log exp
$xpaget ds9 scale datasec
$xpaget ds9 scale limits
$xpaget ds9 scale mode
$xpaget ds9 scale scope
$xpaset -p ds9 scale linear
$xpaset -p ds9 scale log 100
$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
$xpaset -p ds9 scale open
$xpaset -p ds9 scale close
```

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:
$xpaget 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:
$xpaget ds9 single
$xpaset -p ds9 single
```

skyview

Support for SkyView image server at HEASARC.

```
Syntax:
skyview [<object>]
        [name <object>]
        [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
        [size <width> <height> degrees|arcmin|arcsec]
        [save yes|no]
        [frame new|current]
        [update frame|crosshair]
        [survey sdssi|sdssr|sdssg|sdssu|sdssg]
        [open|close]
Example:
```

```
$xpaget ds9 skyview name
$xpaget ds9 skyview coord
$xpaget ds9 skyview size
$xpaget ds9 skyview save
$xpaget ds9 skyview frame
$xpaget ds9 skyview survey
$xpaset -p ds9 skyview m31
$xpaset -p ds9 skyview name m31
$xpaset -p ds9 skyview coord 00:42:44.404 +41:16:08.78 sexagesimal
$xpaset -p ds9 skyview size 60 60 arcmin
$xpaset -p ds9 skyview save yes
$xpaset -p ds9 skyview frame current
$xpaset -p ds9 skyview update frame
$xpaset -p ds9 skyview survey sdssi
$xpaset -p ds9 skyview open
$xpaset -p ds9 skyview close
```

sleep

Delays execution for specified number of seconds. Default is 1 second.

```
Syntax:
sleep [#]

Example:
$xpaset -p ds9 sleep
$xpaset -p ds9 sleep 2
```

smooth

Smooth current image or set smooth parameters.

```
Syntax:
smooth []
       [yes|no]
       [function boxcar|tophat|gaussian]
       [radius <int>]
       [open|close]
Example:
$xpaget ds9 smooth
$xpaget ds9 smooth function
$xpaget ds9 smooth radius
$xpaset -p ds9 smooth
$xpaset -p ds9 smooth yes
$xpaset -p ds9 smooth function tophat
$xpaset -p ds9 smooth radius 4
$xpaset -p ds9 smooth open
$xpaset -p ds9 smooth close
```

source

Source TCL code from a file.

```
Syntax:
source [filename]

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

tcl

Execute one tcl command. Must be enabled via the -tcl command line option.

```
Syntax:
tcl [<tcl command>]

Example:
$echo ’puts "Hello, World"’ | xpaset ds9 tcl
$xpaset -p ds9 tcl ’puts "Hello, World"’
```

tile

Controls the tile display mode.

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

Example:
$xpaget ds9 tile
$xpaget ds9 tile mode
$xpaget ds9 tile grid mode
$xpaget ds9 tile grid layout
$xpaget ds9 tile grid gap
$xpaset -p ds9 tile
$xpaset -p ds9 tile yes
$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:
$xpaget ds9 version
```

view

Controls the GUI.

```
Syntax:
view  [layout horizontal|vertical]
      [info yes|no]
      [panner yes|no]
      [magnifier yes|no]
      [buttons yes|no]
      [colorbar yes|no]
      [colorbar horizontal|vertical]
      [colorbar numerics yes|no]
      [graph horizontal|vertical yes|no]
      [filename yes|no[
      [object yes|no]
      [minmax yes|no]
      [lowhigh 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 layout
$xpaget ds9 view info
$xpaget ds9 view panner
$xpaget ds9 view magnifier
$xpaget ds9 view buttons
$xpaget ds9 view colorbar
$xpaget ds9 view graph horizontal
$xpaget ds9 view filename
$xpaget ds9 view object
$xpaget ds9 view minmax
$xpaget ds9 view lowhigh
$xpaget ds9 view frame
$xpaget ds9 view image
$xpaget ds9 view wcsa
$xpaget ds9 view red
$xpaset -p ds9 view layout vertical
$xpaset -p ds9 view info yes
$xpaset -p ds9 view panner yes
$xpaset -p ds9 view magnifier yes
$xpaset -p ds9 view buttons yes
$xpaset -p ds9 view colorbar yes
$xpaset -p ds9 view graph horizontal yes
$xpaset -p ds9 view filename yes
$xpaset -p ds9 view object yes
$xpaset -p ds9 view minmax yes
$xpaset -p ds9 view lowhigh yes
$xpaset -p ds9 view frame 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 [method xpa|mime]
   [server <url>]
   [internal yes|no]
   [delay #]
   [<url>]
   [connect <url>]
   [disconnect <url>]
   [open|close]
Example:
$xpaget ds9 vo
$xpaget ds9 vo method
$xpaget ds9 vo server
$xpaget ds9 vo internal
$xpaget ds9 vo delay
$xpaget ds9 vo connect
$xpaset -p ds9 vo method xpa
$xpaset -p ds9 vo server "http://foo.bar.edu/list.txt"
$xpaset -p ds9 vo internal yes
$xpaset -p ds9 vo delay 15 # keep-alive delay
$xpaset -p ds9 vo chandra-ed
$xpaset -p ds9 vo connect chandra-ed
$xpaset -p ds9 vo disconnect chandra-ed
$xpaset -p ds9 vo open
$xpaset -p ds9 vo close
```

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 [#] <filename>]
    [append [#] <filename>]
    [open|close]
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 wcsa
$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 reset 3
$xpaset -p ds9 wcs replace foo.wcs
$xpaset -p ds9 wcs replace 3 foo.wcs
$xpaset -p ds9 wcs append foo.wcs
$xpaset -p ds9 wcs append 3 foo.wcs
$cat foo.wcs | xpaset ds9 wcs replace
$cat foo.wcs | xpaset ds9 wcs append
$echo "OBJECT = ’foobar’" | xpaset ds9 wcs append
$xpaset -p ds9 open
$xpaset -p ds9 close
```

web

Display specified URL in the web display.

```
Syntax:
web [new|<webname>] [<url>]
    [<webname>] [click back|forward|stop|reload|#]
    [<webname>] [clear]
    [<webname>] [close]

Example:
$xpaget ds9 web
$xpaset -p ds9 web www.cnn.com
$xpaset -p ds9 web new www.cnn.com
$xpaset -p ds9 web hvweb www.apple.com
$xpaset -p ds9 web click back
$xpaset -p ds9 web click 2
$xpaset -p ds9 web clear
$xpaset -p ds9 web close
```

width

Set the width of the image display window.

```
Syntax:
width [<value>]

Example:
$xpaget ds9 width
$xpaset -p ds9 width 512
```

zscale

Set Scale Limits based on the *IRAF* algorithm.

```
Syntax:
zscale []
       [contrast]
       [sample]
       [line]

Example:
$xpaget ds9 zscale contrast
$xpaget ds9 zscale sample
$xpaget ds9 zscale line
$xpaset -p ds9 zscale
$xpaset -p ds9 zscale contrast .25
$xpaset -p ds9 zscale sample 600
$xpaset -p ds9 zscale line 120
```

zoom

Controls the current zoom value for the current frame.

```
Syntax:
zoom [<value>]
     [<value> <value>]
     [to <value>]
     [to <value> <value>]
     [to fit]
     [open|close]

Example:
$xpaget ds9 zoom
$xpaset -p ds9 zoom 2
$xpaset -p ds9 zoom 2 4
$xpaset -p ds9 zoom to 4
$xpaset -p ds9 zoom to 2 4
```

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

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 m*enu.

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

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 that 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"

$width
$height
$depth

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

```
Syntax:
    $width
    $height
Example:
    echo "$width $height $depth" | $text
```

$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
```

$filedialog

Display the standard file dialog. Substitutes the returned pathname. Argument specifies if an open file or save file dialog is invoked.

```
Syntax:
    $filedialog([open|save])
Example:
    echo "$filedialog(open)" | $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
```

$pan

Substitute current pan location of the particular data file are returned. The default coordinate system is `physical`.

```
Syntax:
    $pan
    $pan(<coordinate system>,<format>)
```

where:

```
    coordinate system = 
[image|physical|detector|amplifier|wcs|wcsa...wcsz]
    sky frame         = [fk4|fk5|icrs|galactic|ecliptic]
    sky format        = [hms|sexagesimal|degrees]
Example:
    echo $pan(fk5,sexagesimal) | $text
```

$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, xyexe`y) 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 coordinate 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
```

$vo_method

Returns the vo method.

```
Syntax:
    $vo_method
Example:
    echo '$vo_method' | $text
```

$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|wcsa...wcsz]
    sky frame         = [fk4|fk5|icrs|galactic|ecliptic]
    sky format        = [hms|sexagesimal|degrees]
Example:
    echo $x(fk5,sexagesimal) $y(fk5,sexagesimal) | $text
```

$xpa

Returns the xpa access point name.

```
Syntax:
    $xpa
Example:
    echo '$xpa' | $text
```

$xpa_method

Returns the xpa method.

```
Syntax:
    $xpa_method
Example:
    echo '$xpa_method' | $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 implemented.

```
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
    *
    menu
    echo $xpa_method | $text
    Test $vo_method
    *
    menu
    echo $vo_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 $xcen $ycen
    *.fits
    menu
    echo "$xcen $ycen" | $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 xyee.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
```

SAMP

SAMP is a messaging protocol that enables astronomy software tools to interoperate and communicate. Broadly speaking, SAMP is an abstract framework for loosely-coupled, asynchronous, RPC-like and/or event-based communication, based on a central service providing multi-directional publish/subscribe message brokering. The message semantics are extensible and use structured but weakly-typed data. For more information on SAMP, please click here.

The samp implementation for DS9 is based on the XPA model with 2 private calls:

```
ds9.get
    Arguments
        cmd (string) required
        url (string) optional
    Returned value
        OK (samp.result map)
            value (string) optional
                url (string) optional
        ERROR (samp.error map)
            samp.errortxt (string)

ds9.set
    Arguments
        cmd (string) required
        url (string) optional
    Returned value
        OK
        ERROR (samp.error map)
            samp.errortxt (string)
```

`ds9.set` maybe called via notification, call and call/wait. `ds9.get` can only be called via call and call/wait. Most of the `ds9.get` calls return a value string, but a few will return a url instead.

`2mass`
`about`
`analysis`
`array`
`bin`
`blink`
`catalog`
`cd`
`cmap`

```
colorbar
console
contour
crosshair
cursor
data
datacube
dsssao
dsseso
dssstsci
exit
file
first
fits
frame
grid
header
height
iconify
iis
imexam
lock
lower
magnifier
mask
match
minmax
mode
nameserver
nvss
orient
pagesetup
pan
pixeltable
plot
prefs
preserve
psprint
print
quit
raise
regions
rgb
rotate
saveimage
savefits
savempeg
```

scale
shm
single
skyview
sleep
smooth
source
tcl
tile
update
version
view
vo
wcs
web
width
zscale
zoom

2mass

Support for 2MASS Digital Sky Survey.

```
Syntax:
2mass [<object>]
      [name <object>]
      [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
      [size <width> <height> degrees|arcmin|arcsec]
      [save yes|no]
      [frame new|current]
      [update frame|crosshair]
      [survey j|h|k]
      [open|close]

Example:
string value = ds9.get(string cmd)
2mass name
2mass coord
2mass size
2mass save
2mass frame
2mass survey
ds9.set(string cmd)
2mass m31
2mass name m31
2mass coord 00:42:44.404 +41:16:08.78 sexagesimal
2mass size 60 60 arcmin
```

```
2mass save yes
2mass frame current
2mass update frame
2mass survey j
2mass open
2mass close
```

about

Get DS9 credits.

```
Syntax:
about

Example:
string url = ds9.get(string cmd)
about
```

analysis

Control external analysis tasks. Tasks are numbered as they are loaded, starting with 0. Can also be used to display a message and display text in the text dialog window.

```
Syntax:
analysis [<task number>]
         [<filename>]
         [task <task number>|<task name>]
         [load <filename>]
         [clear]
         [clear][load <filename>]
         [message ok|okcancel|yesno <message>]
         [entry <message>]
         [text]

Example:
string url = ds9.get(string cmd)
analysis
string value = ds9.get(string cmd)
analysis task
analysis entry Please enter something
analysis entry okcancel Please enter something
ds9.set(string cmd)
analysis 0 # invoke first analysis task
analysis task 0
analysis task foobar
analysis task {foo bar}
analysis my.ans
analysis load my.ans
analysis clear
analysis clear load my.ans
analysis message ok {This is a message}
analysis text {this is text}
ds9.set(string cmd, string url)
analysis load
analysis text
```

array

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

```
Syntax:
array [bigendian|littleendian]
array
[new|mask][[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:
string url = ds9.get(string cmd)
array # default bigendian
array littleendian
ds9.set(string command, string url)
array [dim=512,bitpix=16]
array mask [dim=512,bitpix=16]
array rgb [dim=200,zdim=3,bitpix=8]
array [xdim=512,ydim=512,zdim=1,bitpix=16] # load 512x512 short
array [dim=256,bitpix=-32,skip=4] # 256x256 float with 4 byte head
array [dim=512,bitpix=32,arch=littleendian] # 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>]
    [about center]
    [buffersize <value>]
    [cols <x> <y>]
    [colsz <x> <y> <z>]
    [factor <value> [<vector>]]
    [depth <value>]
    [filter <string>]
    [function average|sum]
    [to fit]
    [open|close]

Example:
string value = ds9.get(string cmd)
bin about
bin buffersize
bin cols
bin factor
bin depth
bin filter
bin function
bin smooth
bin smooth function
bin smooth radius
ds9.set(string cmd)
bin about 4096 4096
bin about center
bin buffersize 512
bin cols detx dety
bin colsz detx dety time
bin factor 4
bin factor 4 2
bin depth 10
bin filter {pha > 5}
bin filter {}
bin function sum
bin to fit
bin open
bin close
```

blink

Blink mode parameters. Interval is in seconds.

```
Syntax:
blink []
      [yes|no]
      [interval <value>]

Example:
string value = ds9.get(string cmd)
blink
blink interval
ds9.set(string cmd)
blink
blink yes
blink interval 1
```

catalog
cat

Support for catalogs. The first three commands will create a new catalog search. All other commands operated on the last search created, unless indicated otherwise.

```
Syntax:
catalog []
        [ned|simbad|denis|skybot]

[ascss|cmc|gsc1|gsc2|gsc3|ac|nomad|ppmx|sao|sdss5|sdss6|tycho|ua2|ub1|ucac2]
        [2mass|iras]
        [csc|xmm|rosat]
        [first|nvss]
        [chandralog|cfhtlog|esolog|stlog|xmmlog]
        [cds <catalogname>]
        [cds <catalogid>]
        [load <filename>]
        [load [xml|sb|tsv] <filename>]
        [<catname>] [samp]
        [<catname>] [samp broadcast]
        [<catname>] [samp send <application>]
        [<catname>] [name <object>]         [<catname>] [coordinate
<ra> <dec> <coordsys>]
        [<catname>] [size <width> <height> degrees|arcmin|arcsec]
        [<catname>] [save <filename>]
        [<catname>] [save [xml|sb|tsv] <filename>]
        [<catname>] [header]
        [<catname>] [filter <string>]
        [<catname>] [filter load <filename>]
        [<catname>] [clear]
        [<catname>] [retrieve]
        [<catname>] [cancel]
        [<catname>] [print]
        [<catname>] [close]
        [<catname>] [server
cds|sao|cadc|adac|iucaa|bejing|cambridge|ukirt]
        [<catname>] [symbol [#]
condition|shape|color|font|fontsize|fontweight|fontslant <value>]
        [<catname>] [symbol [#] text|size|size2|units|angle <value>]
        [<catname>] [symbol shape {circle point}|{box
point}|{diamond point}|
                    {cross point}|{x point}|{arrow point}|{boxcircle
point}|
                    circle|ellipse|box|text]
        [<catname>] [symbol add| [#] remove]
        [<catname>] [symbol save|load <filename>]
        [<catname>] [sort <columnname> incr|decr]
        [<catname>] [maxrows <number>]
        [<catname>] [allrows]
        [<catname>] [allcols]
        [<catname>] [show|hide]
        [<catname>] [ra <columnname>]
        [<catname>] [dec <columnname>]
        [<catname>] [system <coordsys>]
        [<catname>] [sky <skyframe>]
        [<catname>] [show|hide]
        [<catname>] [edit yes|no]
        [<catname>] [panto yes|no]

Example:
string value = ds9.get(string cmd)
catalog
catalog header
catalog cat2mass header
string url = ds9.get(string cmd)
catalog header
ds9.set(string cmd)
catalog
catalog 2mass
catalog cds 2mass
catalog cds {I/252}
catalog load foo.cat
catalog load xml foo.xml
catalog samp broadcast
catalog samp send aladin
catalog cat2mass symbol color red
catalog name m51
catalog coordinate 202.48 47.21 fk5
catalog size 22 22 arcmin
catalog save bar.cat
catalog save xml bar.xml
catalog filter {$Jmag>10}
catalog filter load foo.flt
catalog clear
catalog retrieve
catalog cancel
catalog print
catalog close
catalog server sao
catalog symbol condition {$Jmag>15}
catalog symbol 2 shape {boxcircle point}
catalog symbol color red
catalog symbol font times
catalog symbol fontsize 14
catalog symbol fontweight bold
catalog symbol fontslant italic
catalog symbol add
catalog symbol 2 remove
catalog symbol load foo.sym
catalog symbol save bar.sym
catalog sort {Jmag} incr
catalog maxrows 2000
catalog allrows
catalog allcols
catalog ra RA
catalog dec DEC
catalog system wcs
catalog sky fk5
catalog show
catalog hide
catalog edit yes
catalog panto no
```

cd

Sets/Returns the current working directory.

```
Syntax:
cd [<directory>]

Example:
string value = ds9.get(string cmd)
cd
ds9.set(string cmd)
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 <constrast> <bias>]
     [open|close]

Example:
```

```
string value = ds9.get(string cmd)
cmap
cmap file
cmap invert
cmap value
ds9.set(string cmd)
cmap Heat
cmap file foo.sao
cmap invert yes
cmap value 5 .5
cmap open
cmap close
```

colorbar

Controls colorbar parameters.

```
Syntax:
colorbar [yes|no]
         [horizontal|vertical]
         [orientation horizontal|vertical]
         [numerics yes|no]
         [space value|distance]
         [font times|helvetica|courier]
         [fontsize <value>]
         [fontweight normal|bold]
         [fontslant roman|italic]
         [size]
         [ticks]

Example:
string value = ds9.get(string cmd)
colorbar
colorbar orientation
colorbar numerics
colorbar space
colorbar font
colorbar fontsize
colorbar fontweight
colorbar fontslant
colorbar size
colorbar ticks
ds9.set(string cmd)
colorbar yes
colorbar vertical
colorbar orientation vertical
colorbar numerics yes
colorbar space value
colorbar font times
colorbar fontsize 14
colorbar fontwieght bold
colorbar fontslant italic
colorbar size 20
colorbar ticks 11
```

console

Display tcl console window.

```
Syntax:
-console

Example:
ds9.set(string cmd)
console
```

contour

Controls contours in the current frame.

```
Syntax:
contour []
        [yes|no]
        [<coordsys> [<skyframe>]]
        [clear]
        [generate]
        [load <filename> <coordsys> <skyframe> <color> <width> 
yes|no]
        [save <filename> <coordsys> <skyframe>]
        [convert]
        [loadlevels <filename>]
        [savelevels <filename>]
        [copy]
        [paste <coordsys> <color> <width> yes|no]
        [color <color>]
        [width <width>]
        [dash yes|no]
        [smooth <smooth>]
        [method block|smooth]
        [nlevels <number of levels>]
        [scale linear|log|pow|squared|sqrt|histequ]
        ]scale log exp <value>]
        [mode minmax|<value>|zscale|zmax]
        [limits <min> <max>]
        [levels <value value value...>]
        [open|close]

Example:
string value = ds9.get(string cmd)
contour
contour color
contour width
contour dash
contour smooth
contour method
contour nlevels
contour scale
contour log exp
contour mode
contour limits
contour levels
string url = ds9.get(string cmd)
contour wcs fk5
ds9.set(string cmd)
contour
contour yes
contour clear
contour generate
contour load ds9.con wcs fk5 yellow 2 no # solid line
contour load ds9.con wcs fk5 red 2 yes # dashed line
contour save ds9.con wcs fk5
contour convert
contour loadlevels ds9.lev
contour savelevels ds9.lev
contour copy
contour paste wcs red 2 no
contour color yellow
contour width 2
contour dash yes
contour smooth 5
contour method smooth
contour nlevels 10
contour scale sqrt
contour log exp 1000
contour mode zscale
contour limits 1 100
contour levels "{1 10 100 1000}"
contour open
contour close
```

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 <coordsys> [<skyframe>][<skyformat>]]

Example:
string value = ds9.get(string cmd)
crosshair # get crosshair in physical coords
crosshair wcs fk4 sexagesimal # get crosshair in wcs coords
ds9.set(string cmd)
crosshair 100 100 physical # set crosshair in physical
crosshair 345 58.8 wcs fk5 # set crosshair in wcs coords
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:
ds9.set(string cmd)
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 of yes indicates to strip the coordinates from the output and just list the data values. The default is yes.

```
Syntax:
data [<coordsys> [<skyframe>] <x> <y> <w> <h> [yes|no]]

Example:
string url = ds9.get(string cmd)
data image 450 520 3 3 yes
data physical 899 1039 6 6 no
data fk5 202.47091 47.196811 0.00016516669 0.00016516669 no
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 #]
         [axis #]
         [open|close]

Example:
string value = ds9.get(string cmd)
datacube
datacube interval
ds9.set(string cmd)
datacube play
datacube last
datacube 3
datacube interval 2
datacube axis 3
datacube open
datacube close
```

dsssao

Support for Digital Sky Survey at SAO.

```
Syntax:
dsssao [<object>]
       [name <object>]
       [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
       [size <width> <height> degrees|arcmin|arcsec]
       [save yes|no]
       [frame new|current]
       [update frame|crosshair]
       [open|close]

Example:
string value = ds9.get(string cmd)
dsssao name
dsssao coord
dsssao size
dsssao save
dsssao frame
ds9.set(string cmd)
dsssao m31
dsssao name m31
dsssao coord 00:42:44.404 +41:16:08.78 sexagesimal
dsssao size 60 60 arcmin
dsssao save yes
dsssao frame current
dsssao update frame
dsssao open
dsssao close
```

dsseso

Support for Digital Sky Survey at ESO.

```
Syntax:
dsseso [<object>]
       [name <object>]
       [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
       [size <width> <height> degrees|arcmin|arcsec]
       [save yes|no]
       [frame new|current]
       [update frame|crosshair]
       [survey DSS1|DSS2-red|DSS2-blue|DSS2-infrared]
       [open|close]

Example:
string value = ds9.get(string cmd)
dsseso name
dsseso coord
dsseso size
dsseso save
dsseso frame
dsseso survey
ds9.set(string cmd)
dsseso m31
dsseso name m31
dsseso coord 00:42:44.404 +41:16:08.78 sexagesimal
dsseso size 60 60 arcmin
dsseso save yes
dsseso frame current
dsseso update frame
dsseso survey DSS2-red
dsseso open
dsseso close
```

dssstsci

Support for Digital Sky Survey at STSCI.

```
Syntax:
dssstsci [<object>]
         [name <object>]
         [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
         [size <width> <height> degrees|arcmin|arcsec]
         [save yes|no]
         [frame new|current]
         [update frame|crosshair]
         [survey poss2ukstu_red|poss2ukstu_ir|poss2ukstu_blue]
         [survey poss1_blue|poss1_red]
         [survey all|quickv|phase2_gsc2|phase2_gsc1]
         [open|close]

Example:
string value = ds9.get(string cmd)
dssstsci name
dssstsci coord
dssstsci size
dssstsci save
dssstsci frame
dssstsci survey
ds9.set(string cmd)
dssstsci m31
dssstsci name m31
dssstsci coord 00:42:44.404 +41:16:08.78 sexagesimal
dssstsci size 60 60 arcmin
dssstsci save yes
dssstsci frame current
dssstsci update frame
dssstsci survey all
dssstsci open
dssstsci close
```

exit
quit

Quits DS9.

```
Syntax:
exit
quit

Example:
ds9.set(string cmd)
exit
```

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|mask][<filename>]
     [new|mask][fits <filename>]
     [new|mask][sfits <filename> <filename>]
     [new|mask][medatacube <filename>]
     [new|mask][mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] <filename>]
     [new|mask][mosaic [iraf|wcs|wcsa...wcsz] <filename>]
     [new|mask][smosaic [iraf|wcs|wcsa...wcsz] <filename>
<filename>]
     [new][multiframe <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|mask][array
<file>[[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:
string value = ds9.get(string cmd)
file
ds9.set(string cmd)
file foo.fits
file mask foo.fits
file fits foo.fits
file sfits foo.hdr foo.arr
file medatacube foo.fits
file mosaicimage iraf bar.fits
file mosaicimage wcs bar.fits
file mosaicimage wcsa bar.fits
file mosaicimage wfpc2 hst.fits
file mosaic iraf foo.fits
file mosaic wcs bar.fits
file smosaic iraf foo.hdr foo.arr
file smosaic wcs bar.hdr bar.arr
file multiframe foo.fits
file rgbcube rgb.fits
file srgbcube rgb.hdr rgb.arr
file rgbimage rgb.fits
file rgbarray rgb.arr[dim=200,zdim=3,bitpix=-32]
file array array.arr[dim=512,bitpix=-32]
file url 'ftp://foo.bar.edu/img.fits'
file save foo.fits # save the current frame as FITS Image
file save gz foo.fits.gz # save as compressed FITS Image
file save resample foo.fits # save current pan/zoom/rotate as FITS
Image
file save resample gz foo.fits.gz # save as compressed FITS Image
```

first

Support for VLA First Sky Survey.

```
Syntax:
first [<object>]
      [name <object>]
      [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
      [size <width> <height> degrees|arcmin|arcsec]
      [save yes|no]
      [frame new|current]
      [update frame|crosshair]
      [open|close]

Example:
string value = ds9.get(string cmd)
first name
first coord
first size
first save
first frame
ds9.set(string cmd)
first m31
first name m31
first coord 00:42:44.404 +41:16:08.78 sexagesimal
first size 60 60 arcmin
first save yes
first frame current
first update frame
first open
first close
```

fits

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

```
Syntax:
fits [size|width|height|depth|bitpix]
     [size [wcs|wcsa...wcsz] [fk4|fk5|icrs|galactic|ecliptic]
[degrees|arcmin|arcsecs]]
     [type]
     [header [#] [keyword <string>]]
     [image|table|resample] [gz]
     [new|mask][<options>]
     [new|mask][medatacube <options>]
     [new|mask][mosaicimage [iraf|wcs|wcsa...wcsz|wfpc2] <options>]
     [new|mask][mosaic [iraf|wcs|wcsa...wcsz] <options>]
     [new][rgbcube <options>]
     [new][rgbimage <options>]
     [save resample gz <filename>]

Example:
string value = ds9.get(string cmd)
fits size
fits width
fits height
fits depth
fits bitpix
fits size wcs fk5 arcmin
fits type
fits header keyword "'BITPIX'"
fits header 1 keyword "'BITPIX'"
string url = ds9.get(string cmd)
fits
fits image
fits image gz
fits table
fits table gz
fits resample
fits resample gz
fits header # primary
fits header 2 # hdu 2
fits header -2 # hdu 2 with inherit
ds9.set(string cmd, string url)
fits
fits [2]
fits new [bin=detx,dety]
fits medatacube
fits mosaicimage iraf
fits mosaicimage wcs
fits mosaicimage wcsa
fits mosaicimage wfpc2
fits mosaic iraf
fits mosaic wcs
fits rgbcube
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]]
      [move first]
      [move back]
      [move forward]
      [move last]
      [first]
      [prev]
      [next]
      [last]
      [frameno #]
      [#]
      [has
[amplifier|datamin|datasec|detector|grid|iis|irafmin|physical|smooth]]
      [has contour [aux]]]
      [has fits [ |bin|cube|mosaic]]
      [has marker [highlite|paste|select|undo]]
      [has system <coordsys>]
      [has wcs [<wcssys>|equatorial <wcssys>|linear <wcssys>]]

Example:
string value = ds9.get(string cmd)
frame # returns the id of the current frame
frame frameno # returns the id of the current frame
frame all # returns the id of all frames
frame active # returns the id of all active frames
frame has amplifier
frame has datamin
frame has datasec
frame has detector
frame has grid
frame has iis
frame has irafmin
frame has physical
frame has smooth
frame has contour
frame has contour aux
frame has fits
frame has fits bin
frame has fits cube
frame has fits mosaic
frame has marker highlite
frame has marker paste
frame has marker select
frame has marker undo
frame has system physical
frame has wcs wcsa
frame has wcs equatorial wcsa
frame has wcs linear wcsa
ds9.set(string cmd)
frame center # center current frame
frame center 1 # center ’Frame1’
frame center all # center all frames
frame clear # clear current frame
frame new # create new frame
frame new rgb # create new rgb frame
frame delete # delete current frame
frame reset # reset current frame
frame refresh # refresh current frame
frame hide # hide current frame
frame show 1 # show frame ’Frame1’
frame move first # move frame to first in order
frame move back # move frame back in order
frame move forward # move frame forward in order
frame move last # move frame to last in order
frame first # goto first frame
frame prev # goto prev frame
frame next # goto next frame
frame last # goto last frame
frame frameno 4 # goto frame ’Frame4’, create if needed
frame 3 # goto frame ’Frame3’, create if needed
```

grid

Controls coordinate grid. For grid numeric format syntax, click here.

```
Syntax:
grid  []
      [yes|no]
      [type analysis|publication]
      [system <coordsys>]
      [sky <skyframe>]
      [skyformat <skyformat>]
      [grid yes|no]
      [grid color <color>]
      [grid width <value>]
      [grid style 0|1]
      [grid gap1 <value>]
      [grid gap2 <value>]
      [axes yes|no]
      [axes color <color>]
      [axes width <value>]
      [axes style 0|1]
      [axes type interior|exterior]
      [format1 <format>]
      [format2 <format>]
      [tick yes|no]
      [tick color <color>]
      [tick width <value>]
      [tick style 0|1]
      [border yes|no]
      [border color <color>]
      [border width <value>]
      [border style 0|1]
      [numlab yes|no]
      [numlab font times|helvetica|courier]
      [numlab fontsize <value>]
      [numlab fontweight normal|bold]
      [numlab fontslant roman|italic]
      [numlab color <color>]
      [numlab gap1 <value>]
      [numlab gap2 <value>]
      [numlab type interior|exterior]
      [numlab vertical yes|no]
      [title yes|no]
      [title text <text>]
      [title def yes|no]
      [title gap <value>]
      [title font times|helvetica|courier]
      [title fontsize <value>]
      [title fontweight normal|bold]
      [title fontslant roman|italic]
      [title color <color>]
      [textlab yes|no]
      [textlab text1 <text>]
      [textlab def1 yes|no]
      [textlab gap1 <value>]
      [textlab text2 <text>]
      [textlab def2 yes|no]
      [textlab gap2 <value>]
      [textlab font times|helvetica|courier]
      [textlab fontsize <value>]
      [textlab fontweight normal|bold]
      [textlab fontslant roman|italic]
      [textlab color <color>]
      [reset]
      [load <filename>]
      [save <filename>]
      [open|close]
Example:
string value = ds9.get(string cmd)
grid
grid type
grid system
grid sky
grid skyformat
grid grid
grid grid color
grid grid width
grid grid style
grid grid gap1
grid grid gap2
grid axes
grid axes color
grid axes width
grid axes style
grid axes type
grid format1
grid format2
grid tick
grid tick color
grid tick width
grid tick style
grid border
grid border color
grid border width
grid border style
grid numlab
grid numlab font
grid numlab fontsize
grid numlab fontweight
grid numlab fontslant
grid numlab color
grid numlab gap1
grid numlab gap2
grid numlab type
grid numlab vertical
grid title
grid title text
grid title def
grid title gap
grid title font
grid title fontsize
grid title fontweight
grid title fontslant
grid title color
grid textlab
grid textlab text1
grid textlab def1
grid textlab gap1
grid textlab text2
grid textlab def2
grid textlab gap2
grid textlab font
grid textlab fontsize
grid textlab fontweight
grid textlab fontslant
grid textlab color
ds9.set(string cmd)
grid
grid yes
grid type analysis
grid system wcs
grid sky fk5
grid skyformat degrees
grid grid yes
grid grid color red
grid grid width 2
grid grid style 1
grid grid gap1 10
grid grid gap2 10
grid axes yes
grid axes color red
grid axes width 2
grid axes style 1
grid axes type exterior
grid format1 d.2
grid format2 d.2
grid tick yes
grid tick color red
grid tick width 2
grid tick style 1
grid border yes
grid border color red
grid border width 2
grid border style 1
grid numlab yes
grid numlab font courier
grid numlab fontsize 12
grid numlab fontweight bold
grid numlab fontslant italic
grid numlab color red
grid numlab gap1 10
grid numlab gap2 10
grid numlab type exterior
grid numlab vertical yes
grid title yes
grid title text {Hello World}
grid title def yes
grid title gap 10
grid title fontsize 12
grid title font courier
grid title fontweight bold
grid title fontslant italic
grid title color red
grid textlab yes
grid textlab text1 {Hello World}
grid textlab def1 yes
grid textlab gap1 10
grid textlab text2 {Hello World}
grid textlab def2 yes
grid textlab gap2 10
grid textlab font courier
grid textlab fontsize 12
grid textlab fontweight bold
grid textlab fontslant italic
grid textlab color red
grid reset
grid load foo.grd
grid save foo.grd
grid open
grid close
```

header

Display current fits header dialog. Optional extension number maybe specified. Please note, this differs from xpa fits header.

```
Syntax:
header [<value>]
       [close [<value>]]

Example:
string value = ds9.get(string cmd)
header
header 2
header close
```

height

Set the height of the image display window.

```
Syntax:
height [<value>]

Example:
string value = ds9.get(string cmd)
height
ds9.set(string cmd)
height 512
```

iconify

Toggles iconification.

```
Syntax:
iconify []
        [yes|no]

Example:
string value = ds9.get(string cmd)
iconify
ds9.set(string cmd)
iconify
iconify yes
```

iis

Set/Get IIS Filename. Optional mosaic number maybe supplied.

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

Example:
string value = ds9.get(string cmd)
iis filename
iis filename 4
ds9.set(string cmd)
iis filename foo.fits
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 <coordsys> [<skyframe>] [<skyformat>]]
       [key] [coordinate <coordsys> [<skyframe>] [<skyformat>]]
       [any] [coordinate <coordsys> [<skyframe>] [<skyformat>]]
       [] [data [width][height]]
       [key] [data [width][height]]
       [any] [data [width][height]]

Example:
string value = ds9.get(string cmd)
imexam coordinate image
imexam key coordinate image # return coordinate and key event
imexam any coordinate image # return coordinate and key/mouse event
imexam coordinate wcs fk5 degrees
imexam coordinate wcs galactic sexagesimal
imexam coordinate fk5
imexam data # return data value
imexam key data # return data value and key event
imexam any data # return data value and key/mouse event
imexam data 3 3 # return all data in 3x3 box about selected point
```

lock

Lock frames.

```
Syntax:
lock [crosshair none|wcs|wcsa...wcsz|physical|image]
```

```
Example:
ds9.set(string cmd)
lock crosshair wcs
```

lower

Lower in the window stacking order.

```
Syntax:
lower

Example:
ds9.set(string cmd)
lower
```

magnifier

Controls the magnifier settings.

```
Syntax:
magnifier [color <color>]
          [zoom <value>]
          [cursor yes|no]
          [region yes|no]

Example:
string value = ds9.get(string cmd)
magnifier color
magnifier zoom
magnifier cursor
magnifier region
ds9.set(string cmd)
magnifier color yellow
magnifier zoom 2
magnifier cursor no
magnifier region no
```

mask

Controls mask parameters.

```
Syntax:
mask [color <color>]
     [mark 1|0]
     [transparency <value>]
     [clear]
     [open|close]
```

```
Example:
string value = ds9.get(string cmd)
mask color
mask mark
mask transparency
ds9.set(string cmd)
mask color red
mask mark 0
mask transparency 50
mask clear
mask open
mask close
```

match

Match all other frames to the current frame.

```
Syntax:
match [frames wcs|physical|image]
      [colorbars]
      [scales]
      [bin]

Example:
ds9.set(string cmd)
match frames wcs
match colorbars
match scales
match bin
```

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

Example:
string value = ds9.get(string cmd)
minmax mode
minmax interval
ds9.set(string cmd)
```

```
minmax scan
minmax mode scan
minmax interval 10
```

mode

Controls the first mouse button mode.

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

Example:
string value = ds9.get(string cmd)
mode
ds9.set(string cmd)
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]
           [pan]
           [crosshair]
           [close]

Example:
string value = ds9.get(string cmd)
nameserver
nameserver server
nameserver skyformat
nameserver m31
ds9.set(string cmd)
nameserver m31
nameserver name m31
nameserver server ned-sao
nameserver skyformat sexagesimal
nameserver pan
nameserver crosshair
nameserver open
nameserver close
```

nvss

Support for NRAO VLA Sky Survey.

```
Syntax:
nvss [<object>]
     [name <object>]
     [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
     [size <width> <height> degrees|arcmin|arcsec]
     [save yes|no]
     [frame new|current]
     [update frame|crosshair]
     [open|close]

Example:
string value = ds9.get(string cmd)
nvss name
nvss coord
nvss size
nvss save
nvss frame
ds9.set(string cmd)
nvss m31
nvss name m31
nvss coord 00:42:44.404 +41:16:08.78 sexagesimal
nvss size 60 60 arcmin
nvss save yes
nvss frame current
nvss update frame
nvss open
nvss close
```

orient

Controls the orientation of the current frame.

```
Syntax:
orient [none|x|y|xy]
       [open|close]

Example:
string value = ds9.get(string cmd)
orient
ds9.set(string cmd)
orient xy
orient open
orient close
```

pagesetup

Controls Page Setup options.

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

Example:
string value = ds9.get(string cmd)
pagesetup orientation
pagesetup pagescale
pagesetup pagesize
ds9.set(string cmd)
pagesetup orientation portrait
pagesetup pagescale scaled
pagesetup pagesize poster
```

pan

Controls the current image cursor location for the current frame.

```
Syntax:
pan [x y <coordsys> [<skyframe>][<skyformat>]]
    [to x y <coordsys> [<skyframe>][<skyformat>]
    [open|close]
    [close]

Example:
string value = ds9.get(string cmd)
pan # get current image coords
pan wcs fk4 sexagesimal # get current wcs coords
ds9.set(string cmd)
pan 200 200 image # pan relative
pan to 400 400 physical # pan to physical coords
pan to 13:29:55 47:11:50 wcs fk5 # pan to wcs coords
pan open
pan close
```

pixeltable

Display/Hide the pixel table.

```
Syntax:
pixeltable []
           [yes|open]
           [no|close]
```

```
Example:
string url = ds9.get(string cmd)
pixeltable
ds9.set(string cmd)
pixeltable
pixeltable yes
pixeltable open
pixeltable close
```

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>]]
     [new [name <plotname>] <title> <xaxis label> <yaxis label>
 xy|xyex|xyey|xyexey]
# 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>] [print destination 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|courier]
     [<plotname>] [font numbers|labels|title size <value>]
     [<plotname>] [font numbers|labels|title weight normal|bold]
     [<plotname>] [font numbers|labels|title slant roman|italic]
# edit current dataset
plot [<plotname>] [dataset #]
     [<plotname>] [view discrete|linear|step|quadratic|error yes|no]
     [<plotname>] [color discrete|linear|step|quadratic|error
<color>]
     [<plotname>] [line discrete circle|diamond|plus|cross]
     [<plotname>] [line linear|step|quadratic|error width <value>]
     [<plotname>] [line linear|step|quadratic dash yes|no]
     [<plotname>] [line error style 1|2]

Example:
# return all plotnames
string value = ds9.get(string cmd)
plot
# create new empty plot window
ds9.set(string cmd)
plot
plot new
plot new name foo
# create new plot with data
ds9.set(string cmd, string url)
plot new stdin
plot new name foo stdin
plot new "{The Title}" "{X}" "{Y}" xy
plot new name foo "{The Title}" "{X}" "{Y}" xy
# load additional dataset into an existing plot
ds9.set(string cmd, string url)
plot data xy # plot additional data
plot foo data xy # plot additional data
# edit existing plot
ds9.set(string cmd)
```

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

prefs

Controls various preference settings.

```
Syntax:
prefs [clear]
      [bgcolor <color>]
      [nancolor <color>]

Example:
string value = ds9.get(string cmd)
prefs bgcolor
prefs nancolor
ds9.set(string cmd)
prefs clear
prefs bgcolor black
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:
string value = ds9.get(string cmd)
preserve scale
preserve pan
preserve regions
ds9.set(string cmd)
preserve scale yes
preserve pan yes
preserve regions yes
```

psprint

For MacOSX and Windows, invokes postscript printing. For all others, same as print. Please see print for further details.

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]
      [resolution 53|72|75|150|300|600]

Example:
string value = ds9.get(string cmd)
print destination
print command
print filename
print palette
print level
print resolution
ds9.set(string cmd)
print
print destination file
print command '{gv -}'
print filename foo.ps
print palette cmyk
print level 2
print resolution 75
```

raise

Raise in the window stacking order.

```
Syntax:
raise

Example:
ds9.set(string cmd)
raise
```

regions

Controls regions in the current frame.

```
Syntax:
regions [<filename>]
        [load [all] <filename>]
        [save <filename>]
        [list [close]]
        [show yes|no]
        [showtext yes|no]
        [centroid]
        [centroid auto yes|no]
        [centroid radius <value>|iteration <value>]
        [getinfo]
        [move front]
        [move back]
        [select all]
        [select none]
        [select group <groupname>]
        [delete all]
        [delete select]
        [format ds9|xml|ciao|saotng|saoimage|pros|xy]
        [system image|physical|wcs|wcsa...wcsz]
        [sky fk4|fk5|icrs|galactic|ecliptic]
        [skyformat degrees|sexagesimal]
        [strip yes|no]
        [shape <shape>]
        [color &ltcolor>]
        [width <width>]
        [delim [nl|<char>]]
        [command <marker command>]
        [composite]
        [dissolve]
        [template <filename>]
        [template <filename> at <ra> <dec> <coordsys> <skyframe>]
        [savetemplate <filename>]
        [groups]
        [group <tag>]
        [group <tag> color &ltcolor>]
        [group <tag> copy]
        [group <tag> delete]
        [group <tag> cut]
        [group <tag> font <font>]
        [group <tag> move <int> <int>]
        [group <tag> movefront]
        [group <tag> moveback]
        [group <tag> property <property> yes|no]
        [group <tag> select]
        [copy]
        [cut]
        [paste image|physical|wcs|wcsa...wcsz]
        [undo]
        [include|exclude|source|background|selected]
        [-format ds9|ciao|saotng|saoimage|pros|xy]
        [-system image|physical|wcs|wcsa...wcsz]
        [-sky fk4|fk5|icrs|galactic|ecliptic]
        [-skyformat degrees|sexagesimal]
        [-delim [nl|<char>]]
        [-prop select|edit|move|rotate|delete|fixed|include|source
1|0]
```

```
        [-group <tag>]
        [-strip yes|no]
        [-wcs yes|no]

Example:
string url = ds9.get(string cmd)
regions
regions -format ds9 -system wcs -sky fk5 -skyformat sexagesimal
-prop edit 1 -group foo
string value = ds9.get(string cmd)
regions show
regions showtext
regions centroid
regions centroid auto
regions centroid radius
regions centroid iteration
regions selected
regions format
regions system
regions sky
regions skyformat
regions strip
regions shape
regions color
regions width
regions delim
regions source
regions background
regions include
regions exclude
regions selected
regions groups
ds9.set(string cmd, string url)
regions -format xy -system wcs -sky fk5
regions -format ds9
ds9.set(string cmd)
regions foo.reg
regions -format ciao bar.reg # load as ciao format
regions foo.fits # FITS regions files do not need a format
specification
regions load foo.reg # load foo.reg into current frame
regions load all foo.reg # load foo.reg into all frames
regions load *.reg# expand *.reg and load into current frame
regions load all *.reg # expand *.reg and load into all frames
regions save foo.reg
regions list
regions list close
regions show yes
regions showtext no
regions centroid
regions centroid auto yes
regions centroid radius 10
regions centroid iteration 20
regions getinfo
regions move back
regions move front
regions select all
regions select none
regions select group foo
regions delete all
regions delete select
regions format ds9
regions system wcs
regions sky fk5
regions skyformat degrees
regions delim nl
regions strip yes
regions shape ellipse
regions color red
regions width 3
regions command "circle 100 100 20"
regions composite
regions dissolve
regions template foo.tpl
regions template foo.tpl at 13:29:55.92 +47:12:48.02 fk5
regions savetemplate foo.tpl
regions group foo color red
regions group foo copy
regions group foo delete
regions group foo cut
regions group foo font ’times 14 bold’
regions group foo move 100 100
regions group foo movefront
regions group foo moveback
regions group foo property delete no
regions group foo select
regions copy
regions cut
regions paste wcs
regions undo
```

rgb

Create RGB frame and control RGB frame parameters.

```
Syntax:
rgb  []
     [red|green|blue]
     [channel [red|green|blue]]
     [view [red|green|blue] [yes|no]]
     [system <coordsys>]
     [lock scale|bin|colorbar|slice|smooth [yes|no]]
     [open|close]

Example:
string value = ds9.get(string cmd)
rgb channel
rgb lock bin
rgb lock scale
rgb lock colorbar
rgb lock slice
rgb lock smooth
rgb system
rgb view red
rgb view green
rgb view blue
ds9.set(string cmd)
rgb # create new rgb frame
rgb red # set current channel to red
rgb channel red # set current channel to red
rgb view blue no # turn off blue channel
rgb system wcs # set rgb coordinate system
rgb lock scale yes # lock rgb channels for scaling
rgb lock bin yes # lock rgb channels for binning
rgb lock colorbar yes # lock rgb colorbar channels
rgb lock slice yes # lock rgb slice channels
rgb lock smooth yes # lock rgb smooth channels
rgb open
rgb close
```

rotate

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

```
Syntax:
rotate [<value>]
       [to <value>]
       [open|close]
```

```
Example:
string value = ds9.get(string cmd)
rotate
ds9.set(string cmd)
rotate 45
rotate to 30
rotate open
rotate close
```

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 <filename>
saveimage jpeg [1-100] <filename>
saveimage tiff [none|jpeg|packbits|deflate] <filename>
saveimage png <filename>
saveimage ppm <filename>
saveimage mpeg [1-31] <filename>

Example:
ds9.set(string cmd)
saveimage fits ds9.fits
saveimage jpeg 75 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.set(string cmd)
savefits ds9.fits
```

savempeg

Save all active frames as a mpeg movie.

```
Syntax:
savempeg [<filename>]

Example:
```

```
ds9.set(string cmd)
savempeg ds9.mpg
```

scale

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

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

Example:
string value = ds9.get(string cmd)
scale
scale log exp
scale datasec
scale limits
scale mode
scale scope
ds9.set(string cmd)
scale linear
scale log exp 100
scale datasec yes
scale histequ
scale limits 1 100
scale mode zscale
scale mode 99.5
scale scope local
scale open
scale close
```

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:
string value = ds9.get(string cmd)
shm
ds9.set(string cmd)
shm 102
shm key 102
shm shmid 102 foo
shm fits key 100 foo
shm sfits key 100 101 foo
shm mosaicimage iraf key 100 foo
shm mosaicimage wcs key 100 foo
shm mosaicimage wcsa key 100 foo
shm mosaicimage wfpc2 key 100 foo
shm mosaicimagenext wcs key 100 foo
shm mosaic iraf key 100 foo
shm mosaic wcs key 100 foo
shm smosaic wcs key 100 101 foo
shm rgbcube key 100 foo
shm srgbcube key 100 101 foo
shm rgbimage key 100 foo
shm rgbarray key 100 [dim=200,zdim=3,bitpix=-32]
shm array shmid 102 [dim=32,bitpix=-32]
shm startload # start a multiple load sequence without updating the
display
shm finishload # finish multiple load sequence
```

single

Select Single Display mode

```
Syntax:
single

Example:
```

```
string value = ds9.get(string cmd)
single
ds9.set(string cmd)
single
```

skyview

Support for SkyView image server at HEASARC.

```
Syntax:
skyview [<object>]
        [name <object>]
        [coord <ra> <dec> degrees|sexagesimal] # in wcs fk5
        [size <width> <height> degrees|arcmin|arcsec]
        [save yes|no]
        [frame new|current]
        [update frame|crosshair]
        [survey sdssi|sdssr|sdssg|sdssu|sdssg]
        [open|close]

Example:
string value = ds9.get(string cmd)
skyview name
skyview coord
skyview size
skyview save
skyview frame
skyview survey
ds9.set(string cmd)
skyview m31
skyview name m31
skyview coord 00:42:44.404 +41:16:08.78 sexagesimal
skyview size 60 60 arcmin
skyview save yes
skyview frame current
skyview update frame
skyview survey sdssi
skyview open
skyview close
```

sleep

Delays execution for specified number of seconds. Default is 1 second.

```
Syntax:
sleep [#]

Example:
```

```
ds9.set(string cmd)
sleep
sleep 2
```

smooth

Smooth current image or set smooth parameters.

```
Syntax:
smooth []
       [yes|no]
       [function boxcar|tophat|gaussian]
       [radius <int>]
       [open|close]

Example:
string value = ds9.get(string cmd)
smooth
smooth function
smooth radius
ds9.set(string cmd)
smooth
smooth yes
smooth function tophat
smooth radius 4
smooth open
smooth close
```

source

Source TCL code from a file.

```
Syntax:
source [filename]

Example:
ds9.set(string cmd)
source foo.tcl
```

tcl

Execute one tcl command. Must be enabled via the -tcl command line option.

```
Syntax:
tcl [<tcl command>]

Example:
ds9.set(string cmd)
```

```
tcl puts "Hello, World"
ds9.set(string cmd, string url)
tcl
```

tile

Controls the tile display mode.

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

Example:
string value = ds9.get(string cmd)
tile
tile mode
tile grid mode
tile grid layout
tile grid gap
ds9.set(string cmd)
tile
tile yes
tile mode row
tile grid
tile grid mode manual
tile grid layout 5 5
tile grid gap 10
tile row
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:
ds9.set(string cmd)
update
update 1 100 100 300 400
update now
update now 1 100 100 300 400
update off # delay refresh of the screen while loading files
update on # be sure to turn it on when you are finished loading
```

version

Returns the current version of DS9.

```
Syntax:
version

Example:
string value = ds9.get(string cmd)
version
```

view

Controls the GUI.

```
Syntax:
view  [layout horizontal|vertical]
      [info yes|no]
      [panner yes|no]
      [magnifier yes|no]
      [buttons yes|no]
      [colorbar yes|no]
      [colorbar horizontal|vertical]
      [colorbar numerics yes|no]
      [graph horizontal|vertical yes|no]
      [filename yes|no[
      [object yes|no]
      [minmax yes|no]
      [lowhigh yes|no]
      [frame yes|no]
      [image|physical|wcs|wcsa...wcsz yes|no]
      [red yes|no]
      [green yes|no]
      [blue yes|no]
```

```
Example:
string value = ds9.get(string cmd)
view layout
view info
view panner
view magnifier
view buttons
view colorbar
view graph horizontal
view filename
view object
view minmax
view lowhigh
view frame
view image
view wcsa
view red
ds9.set(string cmd)
view layout vertical
view info yes
view panner yes
view magnifier yes
view buttons yes
view colorbar yes
view graph horizontal yes
view filename yes
view object yes
view minmax yes
view lowhigh yes
view frame yes
view wcsa yes
view red yes
view green no
view blue yes
```

vo

Invoke an connection to a Virtual Observatory site.

```
Syntax:
vo [method xpa|mime]
   [server <url>]
   [internal yes|no]
   [delay #]
   [<url>]
   [connect <url>]
   [disconnect <url>]
   [open|close]

Example:
string value = ds9.get(string cmd)
vo
vo method
vo server
vo internal
vo delay
vo connect
ds9.set(string cmd)
vo method xpa
vo server "http://foo.bar.edu/list.txt"
vo internal yes
vo delay 15
vo chandra-ed
vo connect chandra-ed
vo disconnect chandra-ed
vo open
vo close
```

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 [#] <filename>]
    [append [#] <filename>]
    [open|close]

Example:
string value = ds9.get(string cmd)
wcs
wcs system
wcs sky
wcs skyformat
wcs align
```

```
ds9.set(string cmd)
wcs wcs
wcs system wcs
wcs wcsa
wcs sky fk5
wcs skyformat sexagesimal
wcs align yes
wcs reset
wcs reset 3
wcs replace foo.wcs
wcs replace 3 foo.wcs
wcs append foo.wcs
wcs append 3 foo.wcs
ds9.set(string cmd, string url)
wcs replace
wcs append
wcs open
wcs close
```

web

Display specified URL in the web display.

```
Syntax:
web [new|<webname>] [<url>]
    [<webname>] [click back|forward|stop|reload|#]
    [<webname>] [clear]
    [<webname>] [close]

Example:
string value = ds9.get(string cmd)
web
ds9.set(string cmd)
web www.cnn.com
web new www.cnn.com
web hvweb www.apple.com
web click back
web click 2
web clear
web close
```

width

Set the width of the image display window.

```
Syntax:
width [<value>]
```

```
Example:
string value = ds9.get(string cmd)
width
ds9.set(string cmd)
width 512
```

zscale

Set Scale Limits based on the *IRAF* algorithm.

```
Syntax:
zscale []
       [contrast]
       [sample]
       [line]

Example:
string value = ds9.get(string cmd)
zscale contrast
zscale sample
zscale line
ds9.set(string cmd)
zscale
zscale contrast .25
zscale sample 600
zscale line 120
```

zoom

Controls the current zoom value for the current frame.

```
Syntax:
zoom [<value>]
     [<value> <value>]
     [to <value>]
     [to <value> <value>]
     [to fit]
     [open|close]

Example:
string value = ds9.get(string cmd)
zoom
ds9.set(string cmd)
zoom 2
zoom 2 4
zoom to 4
zoom to 2 4
zoom to fit
zoom open
zoom close
```

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, saoimage, 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 the 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`,`z2zscale` parameter to auto determine `z1,z2. Here` are the `DISPLAY` parameters in question: parameters to set the upper/lower bounds and distribution. You can also use the

```
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.

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

```
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]

Vector
Usage: vector x1 y1 length angle # vector=[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

Projection3d
Usage: projection3d x y radius

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

Epanda
Usage: epanda x y startangle stopangle nangle inner outer nradius
[angle]

Bpanda
Usage: bpanda x y startangle stopangle nangle inner outer nradius
[angle]

Circle Point
Usage: point x y # point=circle [size]
       circle point x y

Box Point
Usage: point x y # point=box [size]
       box point x y

Diamond Point
Usage: point x y # point=diamond [size]
       diamond point x y

Cross Point
Usage: point x y # point=cross [size]
       cross point x y

X Point
Usage: point x y # point=x [size]
       x point x y
```

```
Arrow Point
Usage: point x y # point=arrow [size]
       arrow point x y

BoxCircle Point
Usage: point x y # point=boxcircle[size]
       boxcircle point x y

Composite
Usage: # composite x y angle
```

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:

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

Dash List

Sets dashed line parameters. This does not render the region in dashed lines.

```
Example: circle(100,100,20) # dashlist = 8 3
```

Width

Sets the line width used to render the region.

```
Example: circle(100,100,20) # width = 2
```

Font

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

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

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 Highlite

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

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

Dash

Render region using dashed lines using current `dashlist` value.

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

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
```

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
```

Include/Exclude

The Include/Exclude properties flags 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
```

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 and size. Use the point property to set the point type.

```
Example: point(100,100) # point=diamond 31
```

Default Properties

The default properties are:

```
text={}
color=green
font="helvetica 10 normal roman"
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 a 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 4.0
```

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 roman" edit=1
move=1 delete=1 highlite=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
usingDTM/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 `wcs#` to specify which wcs to use for all wcs references. Valid values are `wcs, wcsa, wcsb, wcsc... wcsz.`

```
Example: 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              # 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
WCS,WCSA-WCSZ           # specify which WCS system to be used
for
                        # linear and sky coordinate systems
```

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
wcsa; fk4; point 202 47
wcsp; linear; point 100 100
```

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'
```

Note that, for regions which accept a rotation angle such as:

```
ellipse (x, y, r1, r2, angle)
box(x, y, w, h, angle)
```

the angle is relative to the specified coordinate system. In particular, if the region is specified in WCS coordinates, the angle is related to the WCS system, not x/y image coordinate axis. For WCS systems with no rotation, this obviously is not an issue. However, some images do define an implicit rotation (e.g., by using a non-zero CROTA value in the WCS parameters) and for these images, the angle will be relative to the WCS axes. In such case, a region specification such as:

```
fk4;ellipse(22:59:43.985, +58:45:26.92,320", 160", 30)
```

will not, in general, be the same region specified as:

```
physical;ellipse(465, 578, 40, 20, 30)
```

even when positions and sizes match. The angle is relative to WCS axes in the first case, and relative to physical x,y axes in the second.

Composite Region

A Composite Region is a region which is a collection of other regions, which share common properties. A composite region is composed of a center point and a rotation angle, of which all its members are rendered in reference to. A composite region is defined by the # composite x y angle declaration followed by a number of regions who are or'd together. A composite region is manipulated as a single region within ds9. A composite region maybe created from the current selection of regions by selecting the Create Composite Region menu option. Likewise, a composite region can be dissolved by selecting the Dissolve Composite Region menu option.

Template Region

A Template Region is a special form of a region which is saved in a special wcs coordinate system WCS0. WCS0 indicates that the ra and dec values are relative to the current WCS location, not absolute. A template region can be loaded at any location into any fits image which contains a valid wcs. For example, a user may create a series of regions, which represent an instrument template. Then, by selecting the Save As Template menu option, a template region saved. The user may now load this templated into any other fits image which contains a valid WCS.

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 loose some information when writing and then reading back from a region file in a format other that DS9. On output, the regions File Format menu or the XPA regions point is used 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

```
Reading into Funtools from DS9 TEXT is ignored
VECTOR is ignored
PROJECTION is ignored
PROJECTION3D is 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 is ignored
  VECTOR is ignored
  RULER is ignored
  COMPASS is ignored
  TEXT is ignored
  PROJECTION is ignored
  PROJECTION3D is ignored
  ELLIPSE ANNULUS is ignored
  BOX ANNULUS is ignored
  PANDA is translated as PIE
  EPANDA is ignored
  BPANDA is ignored
  All properties are ignored
```

SAOimage

```
  All point regions are translated as POINT
  LINE is ignored
  VECTOR is ignored
  TEXT is ignored
  PROJECTION ignored
  PROJECTION3D is ignored
  RULER is ignored
  COMPASS is ignored
  PANDA is ignored
  EPANDA is ignored
  BPANDA is ignored
  All properties are ignored
```

IRAF PROS

```
  All point regions are  translated as POINT
  LINE is ignored
  VECTOR is ignored
  TEXT is ignored
  RULER is ignored
  COMPASS is ignored
  PROJECTION ignored
  PROJECTION3D is ignored
  PANDA is ignored
  EPANDA is ignored
  BPANDA 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
```

File Formats

FITS

DS9 supports FITS images and FITS binary tables. The following algorithm is used to locate and to load the FITS image or table 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 Image

At load time, the user may provide just a filename or a filename along with FITS extension name or number. FITS extension names are case insensitive. When specifying an extention, be sure to quote strings correctly to pass both the shell and DS9 parser.

```
Syntax:
filename
filename[<name>|#]
Example:
$ds9 foo.fits # default load
$ds9 foo.fits[1] # load first extension
$ds9 foo.fits[BCKGRD] # load extension named 'BCKGRD'
```

FITS Binary Table

At load time, the user may provide just a filename or a filename along with FITS extension name or number, and columns names. FITS extension names and columns names are case insensitive. When specifying a filter, be sure to quote strings correctly to pass both the shell and ds9 parser.

```
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]
ext [<name> | #]
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
```

(see Introduction to Filtering for more information)

```
Example:
$ds9 foo.fits # default load
$ds9 foo.fits[1] # load first extension
$ds9 foo.fits[BCKGRD] # load extension named 'BCKGRD'
$ds9 foo.fits[bin=detx,dety] # bin on detx,dety
$ds9 foo.fits[2][bin=rawx,rawy] # load ext 2, cols rawx,rawy
$ds9 foo.fits[bg_events,bin=rawx,rawy] # load ext bg_events,
cols rawx,rawy
$ds9 foo.fits[bin=x,y,pha] # bin on x,y,pi
$ds9 foo.fits[bin=pi] # bin on x,y,pi
$ds9 'foo.fits[ccd_id==3&&energy>4000]' # quoted filter
$ds9 '"foo.fits[ccd_id==3 && energy>4000]"' # double quoted
filter
$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
```

FITS Data Cube

A FITS Data Cube is a FITS image which contains more than 2 axes (NAXES>2). DS9 will automatically detect if a data cube is present and will load all additional images. At the same time, DS9 will display the DataCube dialog box which allows the user to select which 2 image to be displayed.

FITS Multiple Extension as Data Cube

A FITS Multiple Extension Data Cube file 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 Multiple Extension as Multiple Frames

Load a multiple extension FITS file into multiple frames. Please note that files loaded via standard-in or the xpa fits command can not be displayed using this method.

```
Example:
$ds9 -multiframe foo.fits # load multiple extension fits file as
```

```
multiple frames
```

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 a 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 contains a valid WCS.
HST WFPC2	valid HST WFPC2 data cube, consisting of 4 planes, along with a fits ascii table containing wcs information.

```
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 Mosaic Data Cube

A FITS Mosaic Data Cube is a FITS mosaic image which contains more than 2 axes (NAXES>2). DS9 will automatically detect if a mosaic data cube is present and will load all additional images. At the same time, DS9 will display the DataCube dialog box which allows the user to select which 2 image to be displayed.

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 a 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 providethe 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|64|-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
  ’l’ 64-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: `$filename`

```
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
    CDELT1  =         -2.1277777E-4
    CDELT2  =          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
    CDELT1    -2.1277777E-4
    CDELT2    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 `ds9.ini` is available, it is sourced as the last step in initialization. The following directories are searched in order: `./, $HOME, /usr/local/lib, /opt/local/lib`.

TCL

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

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. Therefor 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, smooths the image before calculating contours. As a result, the larger the smoothness, the slower the result.

Preferences

Allows the user to customize the appearance and behavior of the GUI . Please note: some preferences take effect immediately, while others require DS9 to be restarted. Changes to the preferences can be saved by selecting the `Save` button. Use the `Clear Preferences` button to restore default settings.

User preferences are stored in `.ds9.prf`. DO NOT EDIT this file, since it will be deleted or overwritten by DS9. At startup time, DS9 will search for a preferences file in the following directories, in order, `./, $HOME, /usr/local/lib, /opt/local/lib`.

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 `foo`, then DS9 will look for `.foo.prf.` In this manner, the user can have several predefined preference files that are activated by invoking DS9 with a different application names.

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 tick marks, 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, Tick marks, and Border.

Line

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

Font

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

Numeric Formats

The user may specify custom numeric formats for either axes. The format specification can be empty (default) or a print function, based on the selected coordinate system:

```
image
physical
detector
amplifier
wcs linear
wcs equatorial
```

The format specification string to be passed to the C "printf" function (e.g. "%%1.7G") in order to format a single coordinate value.

The Format string supplied should contain one or more of the following characters. These may occur in any order, but the following is recommended for clarity:

"": Indicates that a plus sign should be prefixed to positive values. By default, no plus sign is used.
"z": Indicates that leading zeros should be prefixed to the value so that the first field is of constant width, as would be required in a fixed-width table (leading zeros are always prefixed to any fields that follow). By default, no leading zeros are added.
"i": Use the standard ISO field separator (a colon) between fields. This is the default behaviour.
"b": Use a blank to separate fields.
"l": Use a letter ("h"/"d", "m" or "s" as appropriate) to separate fields.
"g": Use a letter and symbols to separate fields ("h"/"d", "m" or "s", etc, as appropriate), but include escape sequences in the formatted value so that the Plot class will draw the separators as small super-scripts.
"d": Include a degrees field. Expressing the angle purely in degrees is also the default if none of "h", "m", "s" or "t" are given.
"h": Express the angle as a time and include an hours field (where 24 hours correspond to 360 degrees). Expressing the angle purely in hours is also the default if "t" is given without either "m" or "s".
"m": Include a minutes field. By default this is not included.
"s": Include a seconds field. By default this is not included. This request is ignored if "d" or "h" is given, unless a minutes field is also included.
"t": Express the angle as a time (where 24 hours correspond to 360 degrees). This option is ignored if either "d" or "h" is given and is intended for use where the value is to be expressed purely in minutes and/or seconds of time (with no hours field). If "t" is given without "d", "h", "m" or "s" being present, then it is equivalent to "h".
".": Indicates that decimal places are to be given for the final field in the formatted string (whichever field this is). The "." should be followed immediately by an unsigned integer which gives the number of decimal places required, or by an asterisk. If an asterisk is supplied, a default number of decimal places is used which is based on the value of the Digits attribute.

All of the above format specifiers are case-insensitive. If several characters make conflicting requests (e.g. if both "i" and "b" appear), then the character occurring last takes precedence, except that "d" and "h" always override "t".

The default formats are `d.3` for degrees and `hms.1` / `dms.1` / `ldms.1` for sexagesimal.

How it Works

Table of Contents

How DS9 renders an image

Here is a short description on how DS9 decides to paint a pixel a color on the the screen, give an data value... you need a color scale, a contrast/bias pair for the colorscale, clip values for the data, a scale distribution, and finally, the value of the pixel in question.

Step 1. Select a color scale. A color scale is defined as a number of colors (RGB triplets). The number of RGB triplets can vary from just a few to over 200. DS9 contains a number of predefined color scales (Gray, A, B, I8, ...) or the user may load his own color scale.

Step 2. Apply a contrast/bias pair. This step takes the result of step 1 and creates a new array with the contrast/bias applied. The length of the new array will between 200 (for pseudocolor) and 4096 (for truecolor).

Step 3. Calculate the data clip values (low/high data values). The min/max data values may be used or an algorithm may be used to determine the clip data values.

Step 4. Apply the scale distribution. This involves taking the result of step 2, and creating yet another array, this time of size 16384, redistributing the colors, based on the scale algorithm selected (see Scales).

Step 5. Based on your data clip values, and the value of the pixel you have, index into the result of step 4, and you have an index into lookup table (for pseudocolor) and an RGB pair (for truecolor and postscript).

Scales

The `log` function is defined as the following:

$$y = \frac{\log(ax+1)}{\log(a)}$$

as *x* goes from 0 to 1. The user may specify an exponent *a* to change the distribution of colors within the colorbar. The default value of *a* is 1000. Typically, optical images respond well at 1000, IR images as low as 100, and high energy bin tables up to 10000. A value of 10000 closely matches the **`log`** function of SAOImage as defined as the following:

$$y=\frac{e^{-10x}-1}{e^{-10}-1}$$

The `pow` function is defined as the following:

$$y=\frac{a^{x}-1}{a}$$

as *x* goes from 0 to 1. The user may specify an exponent *a* to change the distribution of colors within the colorbar. The default value of *a* is 1000.

The `sqrt` scale function is defined as the following:

$$y=\sqrt{x}$$

as *x* goes from 0 to 1.

The `square` scale function is defined as the following:

$$y=x^{2}$$

as *x* goes from 0 to 1.

The `histogram equalization` scale function distributes colors based on the frequency of each data value.

Smoothing

The user may select one of three types of smoothing kernels. The parameter, *r* or `kernel radius`, is defined as the following:

Boxcar function, where the width = 2*r*+1
Tophat function, where the radius = *r* and the diameter of kernel is 2*r*+1
Gaussian function, defined as:

$$z=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{1}{2}\frac{(x^2+y^2)}{\sigma^2}},\ \sigma=\frac{r}{2}$$

where the mean = 0 and sigma = *r*/2, and the diameter of kernel is 2*r*+1

Contours

The contour algorithm is from an unknown author and originally came from FV. The difference between the two modes are:

`block` : the image is blocked down before the contour is generated
`smooth` : the image is smoothed via a Gaussian kernel before the contour is generated.

`block` mode is faster as the smoothing parameter increases. Inversely, `smooth` mode is much slower as the smoothing parameter increases.

Large Files

There are several factors that determine if DS9 will be able to load a large file.

32 bit OS vs 64 bit OS : to address very large files, you may need to use an 64 bit OS with a 64bit version of DS9. 32bit apps can address up to 4Gb of address space. However, depending on the OS, this limit may be less. Linux for example, the limit appears to be ~3GB (the OS and shared libs eat up a lot of address space). Under 64bit Solaris, 32bit ds9 has a full 4Gb of space. MacOSX appears to have a limit ~3Gb. Under windows, ~2Gb.

Large File Support: is the ability to sequence thru files larger than 4Gb. DS9 is compiled with LFS.

File system: the OS file system must be able to support files larger than 4Gb. Most recent file systems fully support 4GB>.

Memory Management: There are a number of memory management techniques supported in DS9 that will greatly affect the ability and speed of loading large files:

```
$ ds9 foo.fits # uses mmap
$ cat foo.fits | ds9 - # allocates memory
$ xpaset -p ds9 file foo.fits # uses mmap
$ xpaset -p ds9 fits foo.fits # allocates memory
```

Memory Map (`mmap`) is very fast, limit is memory address space (see above). Allocate is very slow, limit is amount of physical memory + swap partition.

Scanning Data: DS9 needs to determine the min and max data values to correctly display your image. For large files, this can take time. You have several options available under the `Scale` menu:

Scan Data - reads all data
Sample Data - samples every x values (much faster)
Use FITS keyword DATAMIN/MAX or IRAFMIN/MAX - great if in the header, bad because they are always wrong.

The best thing to do is to use sample data, and set the data sample between 10 and 100.

Backup and Restore

DS9 now supports Backup and Restore. When a backup is invoked, DS9 will save in a backup save set all files needed to restore DS9 to that state, including geometry, data files, colormaps, catalogs, contours, and regions.

Backup Save Set

A backup save set consists of a text file, called a backup script, and an optional directory, which will contain auxiliary data files needed to restore DS9 to a previous state. The backup file and the auxiliary directory maybe moved across file systems, or even platforms, but must remain together in the same directory.

Image data files

By default, all data image files are save within the backup save set. However, the user has the option, via the Preferences, to only save only an absolute pathname to the data file, and not the data file itself. This option will dramatically reduce the size of a backup save set, but will restrict the usage to a particular file system and platform.

Image files that have been loaded into DS9 via XPA, SAMP, or from URL will always be saved into the save set.

Caveats

There are several caveats in the usage of Backup and Restore. In particular:

```
  Currently, there is no support for masks.
  External Analysis menus will not be saved.
  Plot Tool windows will not be saved.
  IIS frames (IRAF) will not be saved.
  SAMP and XPA sessions will not be saved.
```

In additions, certain complex images which involve multiple load operations will not be saved correctly:

```
  Open Mosaic IRAF Segment
  Open Mosaic WCS Segment
  Open Multi Ext Data Cube
```

And finally, if the image data had been loaded into DS9 via XPA, SAMP, or from a URL, the following complex load operations are not supported:

```
Open Mosaic IRAF Image
Open Mosaic IRAF Segment Open Mosaic WCS Image
Open Mosaic WCS Segment Open Mosaic WFPC2 Open RGB Fits Image
Open RGB Fits Cube Open RGB Array Open Multi Ext Data Cube Open
Multi Ext Multiple Frames
```