



---

*AHELP for CIAO 3.4*

## prism

Context: gui

*Jump to:* [Description](#) [Examples](#) [Parameters](#) [CHANGES IN CIAO 3.0.2](#) [CHANGES IN CIAO 3.0 XPA](#) [ACCESS POINT AND COMMANDS](#) [ANALYSIS MENU, THE SESSION, AND COMMAND-LINE](#) [OPTIONS](#) [Bugs](#) [See Also](#)

---

## Synopsis

Format-independent file browsing GUI

## Syntax

```
prism [-help] [filename]
```

## Description

In the simplest terms, prism is a graphical user interface (GUI) application for file browsing. It lists the extensions of a file and allows the user to view the header and data of each block. However, prism also offers many other analysis features: interactive plots of column data (including plotting errors and histograms), viewing of simple ascii file, table editing, on-the-fly column selection, image display, header editing, and integration with other CIAO analysis tools.

## File formats

Prism will read in any file that is supported by the CIAO data model, and so DM filters and column specifiers (e.g. "ahelp dmsyntax") can be used when specifying a file to view. Prism can also display simple ASCII files, including those in RDB format. It uses the `ascii2fits` script to convert such files to FITS format and then displays the result (this conversion is done automatically). This means that the format for ASCII files must match that supported by `ascii2fits` (see "ahelp `ascii2fits`" for more information on the allowed formats).

## Prism Settings

The Prism tool has a parameter file which can be edited either by using the CIAO command-line tools such as `plist` and `pset` or via the "Preferences..." option of the "Edit" menu. The parameter file is read each time the "Preferences" option is chosen.

## Window Layout

The data display portion of prism is divided into three major sections: the blocks of a data file (top left window), the header of the selected block (top right window), and the data of the selected block (full lower window). A status window across the bottom of the application provides the user with various feedback messages. Leaving the mouse over a button or field for a few seconds will reveal a pop-up "tool tip" that

gives some information about it (as long as the quickhints parameter in prism's parameter file is set to yes).

User commands to prism are issued through the menu bar across the top of the application, from a pop-up menu (accessed by a the right button when the mouse is in the data field), or with keyboard shortcuts. Additionally, prism interacts with other CXC GUI applications via XPA commands, and can take advantage of the session support in CIAO (see "ahelp session").

## Table display

Tables are displayed as columns, with the column name, units (if defined), and data type listed at the top of the data display. A horizontal scroll bar will appear if all the columns will not fit into the window at once. Vector columns – such as the CHIP, DET, and SKY columns in an event file – are not automatically expanded into their components. To see their contents, highlight the vector column using the left mouse button, and then go to the "Navigate" menu entry and select "Expand Column" (this option can also be found in the floating menu accessed by a right mouse click in the data window, or via the "CTRL+X" shortcut).

## Viewing a subset of the table columns

There are two ways of viewing a subset of the columns in a table. The first is to use a DM filter (see "ahelp dmcols") to select the desired columns. The second is to select the required columns in Prism and then chose the "Reopen w/ Selected" option from the "File" menu (or use the CTRL+r shortcut).

## Plotting columns

When a column has been selected it can be plotted using the Visualization menu. If two or three columns have been selected – note that selecting an unexpanded vector column counts as selecting all components of that column – then they can be plotted using the "Interactive Plot..." option. The first two columns are used as the "x" and "y" values of the plot and the third column – if selected – will be used as the symmetric error value for the "y" value.

If a table only contains two columns then they can be plotted without selection; the first column is taken as the "x" value and the second the "y" value. Similarly if it contains only three columns then they will be taken as the "x", "y", and "y error" values.

## Viewing a histogram of a column

The "Histogram" option of the "Visualization" menu allows you to view a selected column as a histogram. When the option is selected a window will be displayed giving you control over the binning used to create the histogram. Selecting the "Apply" button will display the histogram using the chosen settings whilst keeping the "Histogram Preferences" window open, and so allow you to change the binning parameters; the "OK" button will create the histogram but close this window. Note that the rest of the Prism display will be unresponsive whilst the "Histogram Preferences" window is open.

There are two modes for selecting the binning used for the histogram. The default mode ("Specify Binning Parameter") is to use evenly spaced bins covering a given range where the minimum value, maximum value, and number of bins can be changed by the user. These fields are initially populated with the minimum and maximum values of the selected column and the value of prism's "histbins" parameter (50 by default). The alternative mode ("Specify Bin File") allows you to specify the name of an ASCII or FITS file which lists the lower and upper bounds of the bins to use for the binning. Once a file has been specified you are given the option of selecting which columns from the file should be used for the lower and upper bounds: for ASCII files the columns are referred to as "col1", "col2", ... (note that ASCII files are read in using the ascii2fits script and so need to conform to the layout described in "ahelp ascii2fits").

When a table block is opened then a background process is started to calculate the minimum and maximum values of the numeric columns. These numbers are then used to populate the minimum and maximum value fields in the histogram window. The `rangerows` and `rangecols` parameters – described below – can be used to control this calculation since it can take a non-negligible amount of time for large datasets.

If either `rangecols` or `rangerows` is set to 0 then no range calculations are performed, so the minimum and maximum fields will be left blank when the histogram option is selected. If both are set to -1 then the calculations are performed regardless of the number of rows or cols in the block. Otherwise the numbers of rows and cols must be less than the `rangerows` and `rangecols` values for the ranges to be found.

## Exporting table data to ChIPS

The plotting and histogram options allow simple display of the data. For more complex manipulation or display you can export the selected data to a ChIPS session using the "Export data ..." item of the Visualization menu. This takes the currently-selected columns (or the whole file if none are selected) and copies it into a ChIPS session as the `prism_data` S-Lang variable (if no ChIPS session has already been created then one is started). The contents of `prism_data` are the same as if you had executed a `readfile()` command on the file and included any column filters; the following output shows the `prism_data` structure after exporting the time and sky columns of the file `/data/chandra/evt2.fits`:

```
chips> print(prism_data)
_filename      = evt2.fits
_path          = /data/chandra/
_filter        = [2][cols time,sky]
_filetype      = 4
_header        = String_Type[529]
_ncols         = 3
_nrows         = 21823
time           = Double_Type[21823]
x              = Float_Type[21823]
y              = Float_Type[21823]
```

Note that this currently does not work with image data, nor does it take notice of any row selection you might have made in Prism.

## Image display

The default mode for viewing image data is to display a grid of values in which the bottom-left corner represents the lower-left pixel of the image, which matches the display of `ds9`. To revert to the previous behaviour, where the lower-left pixel is displayed in the first row/column of Prism's display, change the "imgdisplay" parameter of Prism to "no". If you change this parameter via Prism's "Preferences" menu then you need to reload the image block for the new value to have effect.

## Editing data in tables and images

To perform cell editing on tables and images, the file must have write permission. CTRL-(left-click) on the desired cell to enter editing mode, then type in the new value. If you have the `quickhints` parameter set then you can find out this information by moving the mouse over a cell and then leaving it alone for a few seconds.

Valid modifications will result in the cell being changed and a message identifying the change will appear in the status box. Limited range, overflow, and type checking are provided; incorrect values will display an error dialog and a message in the status box. In addition, prism will automatically reset an invalid entry back to its previous value. To edit cell values of arrays or vector columns, expand the column (left-click, then CTRL-x) and enter cell editing mode in the data matrix that is displayed.

An undo option exists for cell edits via the "Edit" menu's "Undo Last Cell Edit" option. The undo option treats

cell edits as a stack; that is, they are undone in reverse order (edits 1, 2, 3 will be undone as 3, 2, 1). When the "Undo Last Cell Edit" option is used, the most recently edited cell is changed back to its prior value and the status message box identifies the change.

Cell edits may only be undone while the window displaying the data is still open. If edits are made to a vector column in an expanded window and the window is closed, the edits will be removed from the undo stack. Appending rows or columns flushes edits to the output file and deletes the entries in the cell editing undo stack. In both these cases, the undo option will no longer be available for the changes.

## Adding columns to a table

To append new columns to the open table extension, select the "Append Table Column..." option from the "Edit" menu and a dialog window will appear. Users may specify a scalar or array column type with the radio toggle. There are fields for the column name, data type, units, and description fields; if 'array' is selected, an additional field for the number of elements is provided. Click [OK] or [Apply] to add the column; a HISTORY keyword that column "colname" was added by prism is written to the block's header. Note that the values are not initialized to any specific value, so they may contain garbage.

## Adding rows to a table

To append one or more rows to the open table extension, select the "Append Table Row(s)..." option from the "Edit" menu and a dialog will appear. Enter the number of rows to append and click on [OK] or [Apply]. Note that the values are initialized to zero. A HISTORY keyword stating that the selected number of rows were added by prism is written to the block's header.

## Recently accessed files list

Selecting the "Open List" option from the "File" menu displays a list of (up to) the last 10 accessed files. Selecting any of the specified files loads it into prism again. The file history is stored in a \$HOME/.prism.filelist file, which the tool checks whenever a file is opened. If the file is already in the list, the list is not modified; if not, the file is added to the top of the list. Filter specifications and path names are considered part of the file name.

## Example 1

The following examples are also described (with images) in the [Introductory threads page](#) of the CIAO website.

### Displaying the image of an event file

Select the coordinate columns to view by left-clicking so that they are highlighted. If using a vector column, only a single column needs to be selected. From the Visualization menu, select "Interactive image" and the imager (ds9) will launch and display the data, binned to the value of the 'binfactor' parameter in prism.par; the binning may also be changed using the "Preferences" option under the "Edit" menu.

## Example 2

### Interactive plotting

Select the two columns to view by left-clicking so that they are highlighted; the first column selected will be plotted on the x-axis, the second on the y-axis. If using a vector column, only a single column needs to be

selected. From the "Visualization menu", select "Interactive plot", which will result in the plot of the data, as well as an xterm window with a session of chips running. This window may be used to filter the data and customize the plot; see 'ahelp chips' for more information.

If a third column is selected then it will be used to plot symmetric errors along the y-axis. If only one column is selected then you can view a histogram of its contents using the "Histogram" menu option.

## Example 3

### Inserting a header keyword

From the "Edit" menu, select the "Edit header..." option and a dialog will appear on the screen. There are 3 text fields – the first line contains a field for the keyword name followed by one for the value and the second line is for a comment string. After typing in the desired values, click on [Insert New Keyword]. This will result in the keyword being added to the open header block. The changes will not be saved to disk until another file is loaded (if the changewarn parameter is set to yes – as it is by default – then the user will be queried before this takes place).

## Parameters

name	type	def	min	max
<u>numrows</u>	integer	20	1	
<u>numlines</u>	integer	3	2	50
<u>binfactor</u>	integer	8	0	
<u>histbins</u>	integer	50	1	
<u>rangerows</u>	integer	1000000	-1	
<u>rangecols</u>	integer	50	-1	
<u>sortkeys</u>	boolean	no		
<u>quickhints</u>	boolean	yes		
<u>changewarn</u>	boolean	yes		
<u>showss</u>	boolean	yes		
<u>rawkeys</u>	boolean	no		
<u>imgdisplay</u>	boolean	yes		
<u>showunits</u>	boolean	yes		
<u>launchchips</u>	boolean	yes		

## Detailed Parameter Descriptions

**Parameter=numrows (integer default=20 min=1)**

*Number of rows in matrix*

This parameter controls the number of rows which are loaded into the scrollable display window. If the number of rows selected is larger than the number which can be viewed, a vertical scrollbar is added to the data display window. This value is also used as the increment value when the [Forward] or [Back] buttons are used to traverse through the data.

**Parameter=numlines (integer default=3 min=2 max=50)**

*Number of lines to display in the status window*

This parameter controls the number of lines which are displayed in the status window. New status messages are appended to the bottom of the window. At present, there is no way to clear the status area.

**Parameter=binfactor (integer default=8 min=0)**

*Binning factor to use for event list columns*

This parameter specifies the blocking factor to use when binning event lists into images.

**Parameter=histbins (integer default=50 min=1)**

*Default number of histogram bins*

When creating a histogram of a column, what is the default number of bins that should be used (when specifying the binning parameters rather than loading them from a file).

**Parameter=rangerows (integer default=1000000 min=-1)**

*Row limit for automatic range calculations*

What is the maximum number of rows of a column to search to find the minimum and maximum values of a column? These values are used to fill in the default limits when creating a histogram.

If set to -1 then all rows are used, and if set to 0 then no rows are used (so the Min/Max fields of the histogram window will be left blank when any numeric column is selected).

**Parameter=rangecols (integer default=50 min=-1)**

*Column limit for automatic range calculations*

What is the maximum number of columns in a block for which the automatic range calculation will be run? These values are used to fill in the default limits when creating a histogram.

If set to -1 then all columns are used, and if set to 0 then no columns are used (so the Min/Max fields of the histogram window will be left blank when any numeric column is selected).

**Parameter=sortkeys (boolean default=no)**

*Sort keywords alphabetically?*

This parameter controls whether the header keywords of a block are displayed in the order in which they are read from the file or if they should be displayed in alphabetical order.

**Parameter=quickhints (boolean default=yes)**

*Display tool tips?*

This parameter controls the state of whether or not prism displays tool tips when the user leaves the cursor over a button/field for a few seconds. If set to 'yes', prism will display brief instructions for the various components of the graphical interface.

**Parameter=changewarn (boolean default=yes)**

*Warn user before saving any changes made to the file?*

This parameter specifies whether prism should warn users about changes to the file before exiting. If this parameter is set to 'yes', a dialog will appear asking if the changes should be saved before the application exits.

**Parameter=showss (boolean default=yes)**

*Show the data subspace in the header window*

This parameter specifies whether or not the data subspace for a block should be printed in the header keyword window.

**Parameter=rawkeys (boolean default=no)**

*Display raw keys in the header window*

This parameter specifies whether keywords should be displayed in their raw format or the way that they are interpreted by the CXC datamodel.

**Parameter=imgdisplay (boolean default=yes)**

*Display image cells with lower left corner as (1,1)*

If this parameter is set, Prism displays image data using the "(1,1) as lower left" convention – so resembling the output of ds9 – and fills the display from the upper left hand corner of the image. If set to No, image data are displayed sequentially from row 1 to n – where row 1 appears as the top row in the display.

Table data are always displayed with row 1 as the top row of the display regardless of the setting of this parameter.

**Parameter=showunits (boolean default=yes)**

*Show column units and datatypes in the data table?*

This parameter specifies whether or not the units and datatypes of columns are displayed in prism's data table.

**Parameter=launchchips (boolean default=yes)**

*Launch distinct chips process for interactive plotting?*

This parameter specifies whether the interactive plot command should launch a new process or take over an existing chips session.

## **CHANGES IN CIAO 3.0.2**

### **File corruption**

It was possible – in rare circumstances – for prism to corrupt a file when viewing it due to a bug in the DataModel. For CIAO 3.0 and CIAO 3.0.1 this could happen if the value printed by the following command was not "0":

```
unix% dmlist <filename> header,raw | grep DSREF | grep ":REG" | wc -l
```

This no longer happens in CIAO 3.0.2.

## Histogram plots on Alpha

It is now possible to create histogram plots of a selected column on the Alpha platform.

## Parameters in "Preferences" window

The "Preferences" window – viewable by selecting the "Preferences" option in the "Edit" menu – now has a vertical scroll bar when the window is too small to display all the options.

## Expanding columns on Linux and OS-X

When viewing a vector column – such as the SKY column in an event file – on Linux and OS-X machines it was possible for Prism to think that you had edited a value and so it would ask you if you wanted to create a back-up of the file. This behaviour no longer occurs.

## CHANGES IN CIAO 3.0

### Viewing ASCII data

If given an ASCII file, prism will attempt to convert it to FITS format using the `ascii2fits` script and then display the converted file. This means that prism can be used to browse simple ASCII files, including those in RDB format.

### Displaying error bars and histograms of column data

If three columns of a table are selected for plotting, the third column will be used as the symmetric error for the "y" component. Prism can now create a histogram of a selected column by using the "Histogram" option of the Visualization menu.

### Exporting column data

Selected columns can be sent to ChIPS as a S-Lang variable for more detailed analysis or display.

### Displaying image data

Images are now displayed so that the lower-left corner of the display represents the lower-left pixel of the image, to match the display of `ds9`. Use the `imgdisplay` parameter to change this behaviour.

### Cell editing

The restriction on only being able to add a value that fits within the displayed cell has been removed, and greater precision is now possible when editing double values.

### Change in name of XPA access point

The names of XPA access points has been simplified in CIAO 3.0 as described in "ahelp session".



## XPA ACCESS POINT AND COMMANDS

As with most of the CIAO GUIs, Prism has an XPA access point which can be used to control its behaviour and query its state – e.g. get prism to load a specific file or ask it what columns are selected. This is used by the CIAO session handling ("ahelp session") to provide an integrated analysis environment.

The default name is "prism", with further instances being called "prism2", "prism3", ... (the "--xpa" command-line option can be used to override this naming scheme). Prism understands the following commands when sent via XPA:

```

unix% xpaget prism
file:          Open file or get filename in browser
               options: <filename>
image:         View/image either current or new file
               options:
                 file <filename> <block column1 column2>
                 rectangle <ra dec width height rotation label> |
                 circle <ra dec radius label> |
                 line <start_ra start_dec end_ra end_dec label> |
                 ellipse <ra dec width height rotation label> |
                 point <ra dec label> |
                 polygon <[ra dec] ... label> |
                 multipoints <[ra dec label] ...> |
                 newarea | area <area num> |
                 region ciao|ds9|saotng <physical|image>
moveto:        Select different block/HDU within the file
               options: <block#>
plot:          View/plot either current or newly specified file
               options:
                 file <filename> <column1> <column2> |
                 <column1> <column2> |
                 print <filename>
                 save <filename>
selected:      Return space-delimited list of selected info
               options:
                 rows (returns list of row numbers) |
                 cols (returns list of column names)
exit:          Exit the current application
               options: None
quit:          Synonym for "exit"

```

Note that many of these commands are intended for use by other GUIs, and so may not be of much use to users.

The following code – which makes use of the S-Lang XPA bindings described in "ahelp xpa" – starts prism and then uses the XPA interface to load the file "/data/Chandra/evt2.fits" after applying the DM filter "sky=circle(4000,4000,100)":

```

chips> require("xpa")
chips> !prism &
chips>
chips> xpaget("prism", "file", "/data/chandra/evt2.fits[sky=circle(4000,4000,100)]")
1
chips> print( xpaget("prism", "file") )
/data/Chandra/evt2.fits[sky=circle(4000,4000,100)]

```

Then if you select the TIME and SKY columns, you can get this selection using

```

chips> print( xpaget("prism", "selected cols") )
time sky
chips> xpaget("prism", "exit")
1

```

The final command just tells prism to finish.

## ANALYSIS MENU, THE SESSION, AND COMMAND-LINE OPTIONS

The Analysis menu can be configured using a simple text file which uses the same format as the DS9 Analysis menu; for more details see "ahelp analysis-menu". For details of the session support in Prism see "ahelp session". As described in that file, Prism has a number of command-line options that can be listed by calling 'prism -help'. These options are not needed by most users and include:

Option	Description
-iconic	Starts prism in an iconified/hidden state, if allowed by the window manager.
-nosession	Do not attach this Prism to the current CIAO session.
-toolmenu <file>	Use <file> rather than the system default file for defining the Analysis menu.
-xpa <name>	Sets the XPA access point of Prism to <name>.

## Bugs

See the [bugs page for this tool](#) on the CIAO website for an up-to-date listing of known bugs.

## See Also

*concept*

[session](#)

*gui*

[analysis-menu](#), [ciao.par](#), [ciaoshmem](#), [filtwin](#), [firstlook](#), [gui](#), [peg](#), [taskmonitor](#)

*tools*

[dmcontour](#), [dmimg2jpg](#), [mkoif](#)

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.  
60 Garden Street, Cambridge, MA 02138 USA.  
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:  
<http://cxc.harvard.edu/ciao3.4/prism.html>  
Last modified: December 2006