NAME

imtool – image display server for the SunView environment

SYNOPSIS

imtool [**–fbconfig** *N*] [**–raster** *filename*]
[**–maptype** (**mono** | **linear** | **random** | **crandom**)]
[**–**[**no**]**colorbar**] [**–white** | **–black**] [*frameargs*]

OPTIONS

–fbconfig *N*

Specifies the initial frame buffer configuration (specifying the number and size of the internal frame buffers) for the server. The set of acceptable frame buffer configurations is defined in a private or public *imtoolrc* file (discussed below).

–raster *filename*

Specifies that the server is to start up with the image in the standard Sun raster format file *filename* already loaded into frame buffer 1. The size of the specified raster need not match the size of the frame buffer configuration specified with the *–fbconfig* argument.

–maptype (**mono** | **linear** | **random** | **crandom**)

Specifies the type of greyscale mapping to be used. **mono** means render the image in black and white (shades of gray). **linear** means render the image in linear pseudocolor, with the range of displayed pixel values corresponding to a range of colors from black to blue to green to red to white. **random** means render the image in random pseudocolor, with a random color being assigned to each greyscale value. **crandom** means render the image in random pseudocolor, updating the colortable every second or so while the mouse is in the display window.

–[**no**]**colorbar**

Specifies whether or not a colorbar is to be displayed at the bottom of the display window. The colorbar graphically illustrates the relationship between pixel intensity and displayed color or greylevel, and is particularly useful with pseudocolor, or when windowing the display. The minimum pixel intensity (1) is at the left and the maximum (200) is at the right.

[**–white** | **–black**]

Specifies whether the image is to be displayed on a white (default) or black background.

[*frameargs*]

imtool also takes generic tool arguments, used to set the position, size, etc. of the display window; see *suntools*(1) for a list of these arguments.

DESCRIPTION

Image Display Server

imtool implements a simple image display server for the SunView window environment. The server runs as an independent process, managing the display window and listening for connections on a pseudodevice entry in **/dev**. All communications with the server are via a simple data stream protocol described below.

Prototype Status

The current implementation of the imtool image display server is a limited prototype, and can be expected to change substantially in the future as the prototype continues to evolve. The main capabilities lacking in the current version are flexibility in lookup table control, overlaid graphics and text, miscellaneous functions such as split screen, and a fully interactive interface to applications programs.

Basic Concepts

The display server consists of a set of N **frame buffers** and a single **display window**. A frame buffer is a two dimensional array in memory into which the image to be displayed is loaded, and which is used to refresh the display window when portions of the window are uncovered. In the current prototype display server only 8 bit deep frame buffers are supported, but the number and size of the frame buffers is user configurable.

The display window is literally a window into the image stored in the frame buffer. While the size of the frame buffer is fixed (subject to periodic reconfiguration), the display window may be any size, and the size may be changed at any time without affecting the contents of the frame buffer. Likewise, the position of a small display window upon a larger frame buffer is arbitrary and may be changed at any time by **panning** the window across the image. While there may be multiple frame buffers, there is only a single display window, and only a single image may be displayed at any one time, although a single keystroke suffices to change the frame being viewed.

With few exceptions, all display server functions are both *independent* and *asynchronous*. Hence, one can display one frame while another is being loaded, or even resize or pan the display window and adjust the greyscale mapping of a frame while the frame is being loaded.

The primary function of the display server is to provide image display and interactive image oriented user interface capabilities to a concurrently executing client program via a bidirectional datastream interface. Hence, to make use of the display server for image display one also needs an applications program capable of talking to the server, and sending it image data to be displayed (a rudimentary builtin capability for displaying Sun raster files is however provided). The **display** program in the IRAF **images.tv** package is an example of such a client program.

The Frame Menu

The imtool frame menu provides the following selections:

Frame	Displays the standard SunView frame menu.
Setup	Displays (or hides) the imtool interactive setup panel.
Register	Adjusts the pan offset of all frame buffers to match that of the image currently being displayed. Normally, the individual frame buffers are independently panned.
Blink [**on**|**off**]	Turns frame blink (alternate display of a series of frames) on or off. Alternatively the **ctrl/B** key may be used to manually cycle through the blink frames.
FitFrame	Adjusts the size of the display window to display the entire frame buffer.
NextFrame	Displays the next frame buffer in sequence. Used to cycle through and alternately display all frames (the **alternate** or **ctrl/F** and **ctrl/R** keys may also be used to cycle forward or reverse through the frames).
Gclear	Clear the graphics overlay of the frame currently being displayed.
Iclear	Clear the image, i.e., frame buffer, currently being displayed.
Imcopy	Make a hardcopy of the image window.

The image hardcopy output function is an entry point to the general screen capture facility, discussed in the next section. This is the same facility used by the *gterm* program.

Hardcopy Output

The image hardcopy function produces a "what you see is what you get" bitmap of the rectangular region of the screen occupied by the display window. If the region of interest is partially covered by another window, then the hardcopy will be a picture of a partially covered window. Any interactive adjustment of the grayscale mapping will be reflected in the hardcopy output.

The screen capture software reads out the full memory of the workstation in the region of interest, and in the case of a color workstation, processes the screen pixels through the colortable to produce an image corresponding to what appears on the screen. No full color output options are currently provided, hence the average of the red, green, and blue color values is next computed. If rasterfile output is being generated, the raw pixel values and RGB color table entries are saved directly in the rasterfile, rather than applying the tables in software to produce a monochrome or bitmap image.

Two output options are currently provided, i.e., **Postscript** output suitable for output directly to a laser writer to produce the final graphics hardcopy, or **Sun rasterfile** output. The default action is to output a Postscript program to the device "lw", e.g., the Apple Laserwriter (any 300 dpi Postscript device should do), using the dithering capability of Postscript to produce a pseudogreyscale representation of the 8 bit output image. These defaults may be changed by defining the following environment variables:

R_RASTERFILE

If this variable is defined a Sun rasterfile will be generated, otherwise a Postscript plotfile is generated. The string value of the variable is a *printf* style format string to be used to generate the filename of the rasterfile. If multiple rasterfiles are to be generated, the format string may contain a decimal integer field (e.g., "frame.%d") to be replaced by the *file number* of the current rasterfile. The first file generated will be number zero, with the file number being incremented once for every rasterfile produced. If Postscript plotfile output is desired, the plotfile will be a uniquely named temporary file in **/tmp**. (Postscript output is text and you can read this file if you are curious what it looks like).

R_DISPOSE

The string value of this variable is a *printf* style format string with one string valued subfield to be replaced by the plotfile or rasterfile name, to be used to generate the command used to dispose of the output file. If this variable is not defined and the output file is a Postscript plotfile, the default format string "lpr -Plw -r -s %s" will be used. If the variable is not defined and the output file is a rasterfile, no action is taken. It is the responsibility of the dispose command to delete the output file.

It should only take several seconds to capture the screen and produce the output rasterfile or queue the Postscript job to the printer. The screen is flashed to indicated when the operation has completed (provided the user has not turned off the *visible bell* feature in their SunView defaults startup file).

The Postscript processing time is usually several minutes (of laserwriter time). Since most Postscript printers are interfaced via a serial interface at 9600 baud, data compression is used to reduce the amount of data to be transmitted, and the current bottleneck is the processing speed of the Postscript engine itself, which does all the dithering and coordinate transformations. (This is true for bitmaps, but the data compression algorithm used is not very effective for 8 bit image data, and the serial interface can still be a bottleneck in this case).

The Setup Panel

The setup panel is used to interactively modify imtool options. Two types of selections are provided, *multiple choice* selections, and *push button* selections. Clicking on a multiple choice selection cycles through the choices (left mouse button) or displays the choices as a menu (right mouse button). String valued options are modified by clicking on the old value, rubbing out all or part of the old value if necessary, and then typing in a new value followed by return. If there several string valued fields in a panel, return may be used to cycle through the fields. Clicking on a push button (use the left mouse button) "pushes" the button, causing the action indicated on the button to be executed.

The multiple choice options in the setup panel are the following:

Number of frame buffers

Specifies the number of frame buffers for which space is currently allocated. The number of frame buffers may be changed at any time. If a client program references a frame which does not yet exist, the number of frame buffers will automatically be increased, hence it is not necessary to preallocate space for the frame buffers.

Greyscale mapping

Specifies the method to be used to map pixel intensity values to RGB color intensity values. **mono** specifies that the image is to be rendered in shades of grey. **linear pseudocolor** specifies that the image is to be rendered in pseudocolor, with pixel intensities mapped into the range of colors from black to blue to green to red to white. **random pseudocolor** assigns a random color to each possible pixel greylevel. **continuous random pseudocolor** is like random pseudocolor,

but the colors are changed every few seconds, where the interval between color changes is specified by the **rate** option below.

Rate (sec) for Crandom Option
Specifies the frequency (1 to 32 seconds) with which new colors are to be assigned for the continuous random pseudocolor option.

Background color
Specifies the frame background color, i.e., the color to which the frame will be set when cleared (black or white), or when displaying an image which fills only part of the frame, the color of the portion of the frame buffer which has not been written into.

Include Frame Border in Imcopy
Specifies whether the frame border, including the frame label, is to be included in image hardcopies. By default the frame border is included since the frame label is often used to identify the displayed image. If the frame border is excluded then so is the colormap, if any.

Show colorbar
Specifies whether or not a colorbar is to be shown at the bottom of the display window.

Blink rate (sec)
When blink is enabled, specifies the amount of time a single frame is to be displayed. The value may range from 1/2 second to 32 seconds. The maximum blink rate may be achieved by holding down the **alternate** key long enough to cause autorepeat.

Frames to be blinked
The string value of this parameter specifies the list of frames to be blinked. The special value "all" may be specified to blink all frames in sequence. If the frames are explicitly listed, the same frame may appear in the list multiple times.

Zoom factors
Specifies the positive integer zoom factors to be used for the zoom and pan function. Up to eight zoom factors may be specified (more would be undesirable due to the time required to cycle through the zooms to get back to an unzoomed image). The default zoom factors are "1 2 4 8"; depending upon the application, one might want to add, e.g., a zoom factor of 3, or a large zoom factor of 32 or 64 to make it easy to identify individual pixels.

Coordinate list output file
Specifies the name of the file to be used for cursor lists (text files containing lists of object coordinates - see below).

Raster filename (load/save)
Specifies the filename of the Sun rasterfile to be loaded into the current frame buffer (**load** push button, below), or the filename into which to current frame is to be written (**save** push button).

The following "push buttons" are also provided in the setup panel. Many of these are equivalent to the comparable selections in the frame menu.

Register Frames Adjust the pan offset of all frames to match that of the current frame.

Fit Window Adjust the size of the display window to match that of the frame buffer.

Reset Reset the display. The mapping type, transfer function, and pan offsets are restored to their initial values, but the contents of the frames are not affected.

Iclear Clear the frame buffer currently being displayed.

Gclear Clear the graphics overlay (may also be cleared whenever the display window is refreshed).

Load Load the Sun rasterfile named by the *Raster filename* panel string parameter into the frame buffer currently being displayed. If the rasterfile is smaller than the frame buffer the raster will be loaded into the upper left corner of the

frame buffer. If the rasterfile is larger than the frame buffer part of the raster will be discarded. Any colortable information present in the rasterfile is ignored.

Save Save the contents of the frame buffer currently being displayed in the Sun rasterfile named by the *Raster filename* panel string parameter. The new raster will be the same size as the frame buffer. No colortable, WCS, or any information other than the pixel values is saved in the rasterfile.

Blink Turn frame blink on or off.

Frame Display the next frame in sequence. Equivalent to the **NextFrame** frame menu selection, and to the **alternate** and **ctrl/F** keys.

Quit Close the setup panel.

Function Keys

The following function keys have special significance to *imtool*:

F4 Calls up the setup panel, or closes it if already displayed.

F5 Causes the current cursor list file to be rewound and reread, marking all objects on the cursor list by drawing a number beside each object in the display window. Each object is marked by its ordinal number in the cursor list, ignoring comment lines and blank lines. The precise object position is at the lower left corner of the first digit. Each time the cursor list is redrawn the color of the digits toggles between black and white, making it possible for the user to manually "blink" the object numbers, or select the representation which provides the best visibility for their data. Note that objects are marked only in the display window, i.e., the frame buffer is not modified, hence the numerals will be lost whenever the display is refreshed.

F6 Enables and disables **cursor readout mode**. While cursor readout mode is in effect a box is displayed in the lower right corner of the display window, in which the coordinates and corresponding pixel intensity at the position of the the mouse cursor are continuously updated as the mouse is moved. The cursor coordinates are given in *world coordinates* if a WCS (world coordinate system) has been defined for the frame, otherwise display window relative pixel coordinates and display pixel intensity values are given. If the pixel intensity is saturated (set to the extreme high or low value), a + or - is appended to the printed value to flag the value as saturated.

F7 Hitting this key while the mouse is in the display window causes an image hardcopy to be generated. This is equivalent to selecting the **imagecopy** item in the frame menu, except that it can be done without moving the mouse. This may be important to avoid changing the greyscale mapping, which also depends upon the mouse position.

Note that the mouse must be in the display window for these function keys to have any effect.

Mouse Buttons

The mouse buttons are used with *imtool* as follows:

Left Button In cursor readout mode, used to mark objects, adding each object to the cursor list for the current frame. Ignored when not in cursor readout mode.

Middle Button The middle button on the mouse is the **pan/zoom** button. If the pan button is held down and released at a position in the display window, the object under the cursor will be moved to the center of the display. **control-pan** is the same

except that the image will pan smoothly to the new position, rather than all at once. **shift-pan** causes the image to be panned in the indicated direction in large steps. Shift and control may be combined to smoothly pan in large steps.

The **zoom** function is also controlled by the middle mouse button. Placing the mouse on an object and pressing pan/zoom once causes the object to be moved to the center of the display; pressing the button again causes the image to be zoomed about the mouse position. Repeated presses without moving the mouse cycle through the predefined set of zoom factors until the cycle wraps around and the unzoomed image is restored. Zoom is almost as fast as a normal unzoomed window refresh, so there is no problem with, for example, panning on a zoomed image.

If the middle button is held down while in cursor readout mode the mouse may be moved without updating the displayed cursor coordinates. This is useful when moving the mouse to a different window, e.g., to type the displayed object coordinates into an application running in some other window. An attempt to pan while already at the edge of the frame is ignored.

Right Button — Used to interactively adjust the greyscale mapping (colortable) for the window.

To window the display, i.e., adjust the **transfer function** for the window, hold the right mouse button down and move the cursor about within the window. Zero contrast (one greylevel) is at the center of the window, with positive contrast above, negative contrast below, and contrast increasing the further the mouse is moved from the centerline. Moving the mouse to the left or right adjusts the greyscale range to lower or higher intensities. The colorbar provides a graphic display of the effect of the transfer function. If the right mouse button is pressed and then released without moving the mouse the transfer function will be adjusted according to the position of the mouse. By alternately displaying several frames and tapping the right mouse button for each frame without moving the mouse, the transfer functions of several frames may be matched.

Frame Buffer Configuration Files

While the prototype display server does support dynamic reconfiguration of the frame buffers, allowing multiple frame buffers of virtually any size, it is currently necessary to define the possible frame buffer configurations at startup time. This is done via a table file called the **imtoolrc** file. An example illustrating the contents of such a file is shown below. Note that this may differ from the default configuration file used at your site.

```
 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 4096 1024        # imt7|imt4x1
 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

20  2  388  576        # imt20|imtgec        GEC CCD detector format
21  1 3040  976        # imt21|imtkpca       KPCA detector format (also 2D-Frutti)
22  1  128 1520        # imt22|imt2df1       2D-Frutti
```

```
23  1  256 1520      # imt23|imt2df2      2D-Frutti
24  1  512 1520      # imt24|imt2df5      2D-Frutti
25  1  960 1520      # imt25|imt2df9      2D-Frutti
26  1  512  800      # imt26|imtcryo      Cryogenic Camera
27  1  348  800      # imt27|imtgcam      Gold Camera
28  1  976 3040      # imt28|imt2df9x3    2D-Frutti
```

Each entry in the file contains four numbers, the configuration number (e.g., as used in *–fbconfig N*), the number of frames to be created initially (frames may be added or deleted once the configuration is specified), and the width and height of the frame in screen pixels. Blank lines and comments are ignored. Note that *the frame width must evenly divisible by 4*, due to alignment restrictions on memory pixrects in SunView.

Selection of the frame buffer configuration to be used is done at run time by the client application program when a frame is loaded. Since the frame buffer size to be used is controlled by the client application program rather than by the server, there is no entry in the setup panel for changing the configuration. Rather, it is expected that a command will be provided at the applications level for specifying the frame buffer size to be used. In the case of IRAF, this is done with a command such as **reset stdimage = imt800**, where the logical device name used on the right is given in the comments in the configuration table above, and must correspond to an equivalent entry in the IRAF **graphcap** file. The imtoolrc file may be customized by the user for special applications, if desired, but a custom version of the graphcap file will be required too.

The frame buffer configuration file may be located in a default public directory, e.g., **/usr/local/lib/imtoolrc**, or the user may have a private version of the file, e.g., **.imtoolrc**. During startup, imtool looks first for an environment variable IMTOOLRC defining the pathname of the imtoolrc file, then it checks for a .imtoolrc file in the user's login directory, and finally it checks for the file /usr/local/lib/imtoolrc. If none of these are found, a default configuration of a single 512 square frame buffer is used.

World Coordinate Systems

imtool provides a simple mechanism for associating a linear **world coordinate system** (WCS) with the displayed image. A WCS consists of an image title string to be displayed in the frame label, a rotation matrix expressing the translation from window pixel coordinates (zero-indexed, origin in the upper left corner of the display window) to *world coordinates*, e.g., the image pixel coordinates of the displayed image, and a pair of image intensity values defining the transformation between display server pixel intensity units (range 1-200 currently) and image pixel intensity units.

The WCS for a frame is passed to imtool as a set-WCS command in the datastream input from the client process (applications display program). A sample WCS descriptor is shown below.

```
dev$pix - m51  B  600s
1. 0. 0. -1. 1. 512. 0. 1481.635 1
```

The first line is simply a line of text to be displayed in the frame label when the WCS is read. This should normally contain the name of the image and a few words describing the image being displayed.

The first six numbers in the second line define a rotation matrix specifying the translation from window pixel coordinates to world coordinates, according to the following relations:

```
x' = ax + cy + tx
y' = bx + dy + ty
```

The coefficients of the transformation matrix are given in the order **a b c d tx ty**. The example defines the image pixel coordinates for a 512 square image, displayed with the origin at [1,1] in the lower left corner of the display window. Note that the world coordinates are flipped in Y and shifted to an origin at [1,1].

The final three numbers on the second line define the transformation from display pixel intensity units to image pixel intensity units. The first two values specify the image pixel intensities corresponding to display pixel intensities 1 and 200. The third number is an integer defining the type of transformation used; currently the value must be 1, indicating a linear transformation.

The WCS is initially undefined when a new frame is created. The first cursor or function key event thereafter which attempts to use WCS information will cause the WCS information to be interpreted if it has been passed in via the set-WCS datastream command. You will know if the server succeeds in reading the WCS because the **frame label** will change when the WCS is read. The first field of the frame label is the frame name, which will be the filename of any cursor list files created for the frame.

Coordinate List Files

Cursor readout mode provides a convenient means of displaying the cursor coordinates, but the coordinates are lost as soon as the cursor is moved. To permanently record the positions of objects of interest one may generate a *coordinate list file* by marking objects with the mouse. The contents of existing list files may also be displayed, and one may append to an existing list. List files generated by foreign programs may easily be displayed.

A coordinate list file is a simple text file with the X-Y coordinate pairs of objects recorded on successive lines in the file. The object number is not recorded explicitly, but is determined by the relative position of a coordinate pair within the file, ignoring comment lines (#...) and blank lines. List files may be edited and redisplayed if desired, and although the object numbers may change they will always agree with whatever is shown on the screen. A hardcopy of the screen may be made to provide a pictorial record of the contents of a list file, e.g., so that objects may be referred to by number if the list is used as input to another program.

The filename of the list file associated with the displayed frame is shown in the frame label, e.g., "frame.1". By default the list file will be created in the WCSDIR, i.e., in the directory defined by the user environment variable WCSDIR.

These defaults may be overridden by entering via the setup panel the *printf* style format string to be used to construct the filename of the list file. If the name given begins with a / an absolute pathname is assumed and the filename is not modified in any way. Otherwise, a directory prefix is prepended, e.g., "$wcsdir/frame.*N*". The directory prefix is specified by the value of the WCSDIR environment variable if defined, otherwise the user's UNIX login directory is assumed. The frame number will appear in the generated filename only if a numeric format is included in the format string, e.g., **frame.%d** to specify a frame number file extension.

Colortable Usage

To permit simultaneous viewing of both the display window and any other windows under SunView, *imtool* uses only a portion of the 256 element hardware colortable. Image pixels range in value from 1 to 200 (colortable entry 0 is reserved for the background or foreground color by SunView). Additional colortable entries are used for the cursor color and a set of graphics overlay colors, reserving 50 or so colortable entries for use by other windows. The graphics colortable entries are in the range 202-217. These are assigned predefined colors, which are unaffected by windowing the display. The graphics colortable assignments are summarized below.

202 black	206 blue	210 coral	214 orchid
203 white	207 yellow	211 maroon	215 turquoise
204 red	208 cyan	212 orange	216 violet
205 green	209 magenta	213 khaki	217 wheat

When imtool is started it reads the hardware color table, which is assumed to contain the entries for the other windows on the screen, replaces entries 1 through 200+, and uses the resultant table to update the hardware colortable thereafter at a rate of twice a second. It is necessary to continuously update the hardware colortable to permit image display while the mouse is not in the display window, however, this scheme will cause colortable conflicts f one attempts to run a second window tool which also uses a large number of colortable entries.

Hints for Blinking Frames

Blink is most effective when used to compare two or more frames which are very similar, e.g., a frame and an edited version of the same frame. For blink to be most effective the images should be displayed as similarly as possible. Start by loading the two frames using the same spatial and greyscale transformation for both frames (IRAF users should use the **repeat** option to the *display* task). Select one of the frames and adjust the window size, pan offset, and greyscale mapping as desired. After adjusting the greyscale with the mouse, repeatedly hit the **alternate** key followed by the right mouse button to match the greyscale mapping of the first frame to the remaining frames. Then select **register** in the frame menu or setup panel to register all the frames, followed by **blink** to blink the frames. Blink works best if the display window is not too large, e.g., a 400 or 500 pixel square window is fine.

Data Stream Protocol

The display server is started like any other SunView tool by executing the *imtool* command either directly or indirectly via the **.suntools** file or via the mouse from the **rootmenu**. When first started a checkered test pattern is displayed, and the display server opens the pseudodevice files **/dev/imt1o** (used by the client to send commands and data to the display server) and **/dev/imt1i** (for data read back from the display server) and begins listening for commands from client programs. All commands and data are passed to the display server as a data stream by applications writing directly to /dev/imt1o.

The data stream protocol used in the prototype server (this will change) mimics that of the IIS Model 70 image display, with an extension added for specifying the frame buffer configuration to be used. The server will automatically sense if the datastream is byte swapped, allowing use of the server with, for example, IRAF running on a large VAX compute server. Only byte packed data is supported.

Note that any process may write to the server, but if more than one process writes to the server at the same time, the output may be garbled. Also, multiple display servers may be spawned, but since only a single pseudodevice entry is currently supported, all such servers would be trying to read from the same input stream.

SEE ALSO

suntools(1), gterm(1), images.tv.display(IRAF)
Windows and Window-Based Tools: Beginner's Guide

ENVIRONMENT

IMTOOLRC The full pathname of the *imtoolrc* (frame buffer configuration) file to be used (optional).

WCSDIR The full pathname of the user directory into which imtool will write any files it creates that are not specified by a full pathname (alias **wcsdir** also permitted).

R_RASTERFILE If defined, a Sun rasterfile rather than Postscript file is generated by the *imcopy* function. The string value is a format used to generate the filename of the rasterfile.

R_DISPOSE The command to be executed to dispose of the Sun rasterfile or Postscript output file created by *imcopy*. If not defined and Postscript output is indicated (R_RASTERFILE also not defined), then the Postscript output file is disposed of via *lpr* to device *lw*.

FILES

/dev/imt1[io]
$WCSDIR/frame.*
/usr/bin/suntools
/usr/lib/rootmenu
$iraf/local/sun/imtool.c

BUGS

(1) The display server is continuously updating the hardware colortable even while the mouse is not in the display window. This is necessary to be able to see the image while the mouse is not in the window, but may interfere with other windows which also use many colortable entries. Under normal circumstances (only one greyscale window) this should not be a problem. If it is

a problem, close the display window when not in use; updating of the hardware colortable is disabled while the display window is closed.

(2) Only a single display server should be used at any one time. Only a single client process should write to the server at any one time.

(3) The environment variables affecting IMTOOL operation must be defined before starting suntools if they are to have any effect.

(4) An error message is printed if imtool encounters a bad data header checksum, e.g., if synchronization is lost on the input datastream or if illegal data is input. Recovery from such an error can be difficult, possibly requiring killing the imtool and starting a new one. This error should not occur if the software (i.e., the client applications program) is functioning properly.

AUTHOR

Doug Tody, National Optical Astronomy Observatories (NOAO), IRAF project.