

dtdisplay

SBC01-????-00

April 15, 1996

Prepared by: J.W. Pflugrath
Molecular Structure Corporation
3200 Research Forest Drive
The Woodlands, Texas 77381
(713) 363-1033
(713) 364-3628 FAX
jwp@msc.com

Prepared for: Contract No. 943072401
Mary Westbrook
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439
(708) 252-8914
(708) 252-4021 FAX
westbrook@anlel.el.anl.gov

Reviewed by:

Approved by:

Copyright © 1995 Molecular Structure Corporation
RESTRICTED RIGHT NOTICE SHORT FORM (JUNE 1987)

Use, reproduction, or disclosure is subject to restrictions set forth in Contract No.
W-31
109-ENG-38 and Contract No. 943072401 with the University of Chicago,
Operator of
Argonne National Laboratory

TABLE OF CONTENTS

1 SCOPE AND PURPOSE	3
2 BACKGROUND	3
3 RUNNING DTDISPLAY	4
3.1 Menu bar	6
3.2. Main window controls.....	19
3.3. Image display areas	21
3.4. Examples	23
4 OVERCOMING DIFFICULTIES	24
4.1 Problems with X Windows.....	25
4.2 Problems with dtdisplay	25
5 INSTALLATION	26
5.1 Application defaults file.....	26
5.2 Make.....	37

1 Scope and Purpose

This document describes how to use the dtdisplay module of d*TREK. dtdisplay displays d*TREK compatible images on a color X Windows display. Since d*TREK images usually depict single crystal diffraction data, dtdisplay has a variety of features that help analyze images within the context of a single crystal diffraction experiment.

With dtdisplay users can:

- Display d*TREK images in any of eight orientations with a variety of color scales,
- Zoom, Pan, and Resize the image display,
- Pick spots and display pixel value, resolution, integrated intensity, and background average,
- Measure the d-spacing between selected spots,
- Display saturated pixels in selected colors.
- Display image headers,
- Plot reflection lists in selected colors,
- Tile a series of images to create a new image,
- Overlay a series of images to create a new image,
- Display images with non-square pixels in true aspect ratio,
- Display images as they are collected by **dtcollect**.

As time progresses, additional features will be incorporated into dtdisplay.

In order to perform certain calculations (e.g. resolution), dtdisplay requires knowledge of the source, the detector and the detector goniometer. dtdisplay gleans all this information from the header of an image file, so that additional input is usually not required.

Definitions of an **image** and a **scan** are found in the Appendix.

2 Background

When you collect and process single crystal diffraction images from a position-sensitive detector, it is very useful to visualize or see the images. You want to check that the crystal diffracts and that the Bragg peaks can be integrated. The suitability of a crystal for further data collection and analysis depends on a number of factors and compromises. Among these factors are:

- crystal unit cell
- spot shape (no split or satellite peaks)
- diffraction limit
- spot separation
- exposure time
- rotation angle increment
- detector position

For example, if the unit cell lengths are large, then the diffraction spots will be spaced close together and the detector will have to be moved further from the crystal. Often only by viewing images can you judge whether or not you should collect diffraction data from a crystal. The dtdisplay program displays images and allows you to decide whether to proceed.

dtdisplay uses a color X Window display and the OSF/Motif widget set and style. Both X and OSF/Motif are widely used on workstations, so you may already be familiar with them and consequently already know how to run and use dtdisplay. If you are not familiar, do not worry as dtdisplay is very easy to use and comes with many help features. Before you start dtdisplay be sure your DISPLAY environment variable is set to your current display and the Dtdisplay application defaults file is accessible to your X server. The latter file can be placed in the system-wide application defaults directory (e.g. /usr/lib/X11/app-defaults) or in your home directory. Please consult the installation guide in section 5 and your system manager.

3 Running dtdisplay

After dtdisplay has been installed and placed in your PATH, to run it just enter dtdisplay along with an optional image filename. Since dtdisplay is an X Window program the standard X command line options are also supported. Messages are written to stdout and stderr as required. The syntax for running dtdisplay is simply:

```
dtdisplay [image_file]
```

Remember, the image filename after the dtdisplay command is optional.

The dtdisplay main window shown below in Figure 1 will appear on the display. The main features of the window are a title, a menu bar with some command buttons, a scrolled text output region, text fields to enter the detector position,

and two areas for image displays. Each of these features is discussed in the following sections.

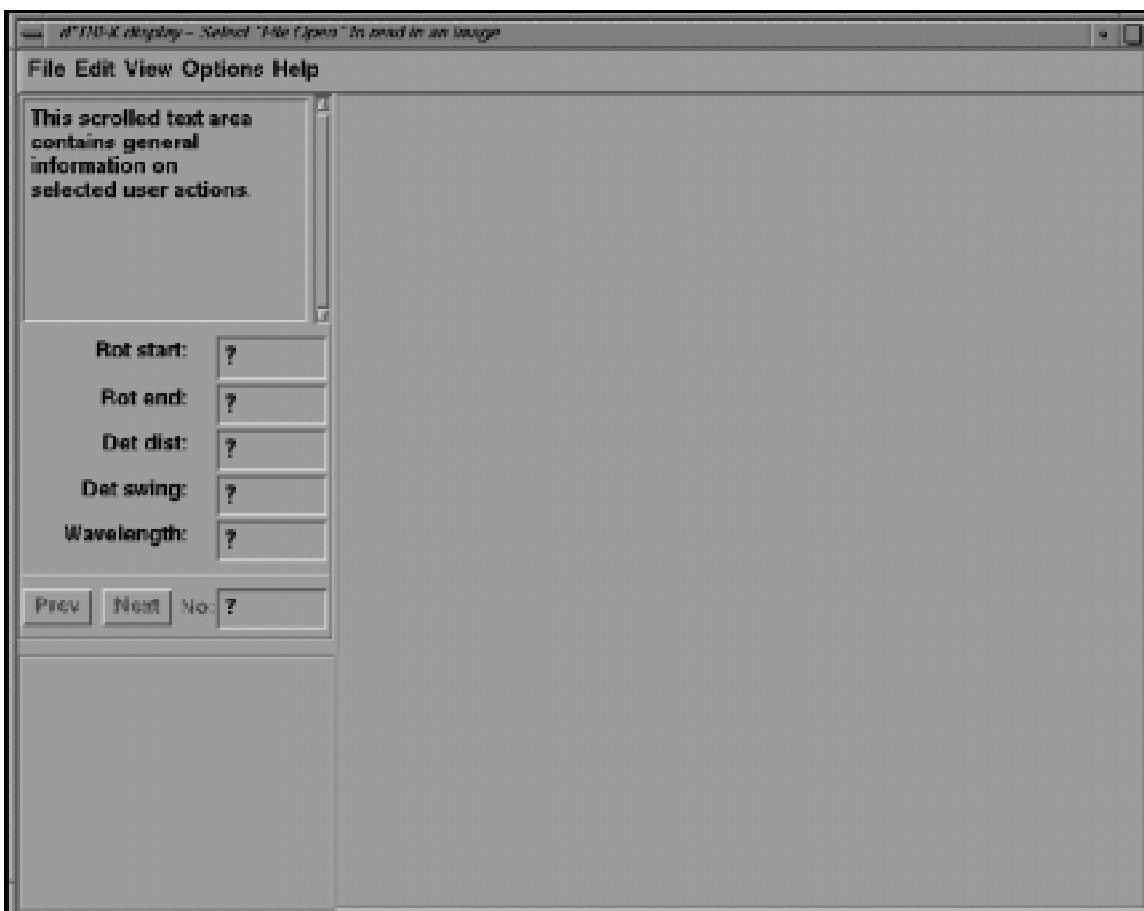


Figure 1. dtdisplay main window before loading an image file.

Getting help

The main window title is usually supplied by the window manager. dtdisplay changes the title in response to user input in two ways. First, when an image is displayed, the title is changed to show the name of the image. Second, when the mouse is used to "arm" a button, the title is changed to show what that button does. In this way, you can receive continuous context-sensitive help. If you ever have a question about what a button does, move the mouse pointer over the button, then press and hold down the mouse button. The title will show what the function of the button is. To activate the function, simply release the mouse button. Do not release the mouse button with the pointer in the button

unless you want to activate the function. To avoid activating the function, move the mouse pointer off the button, then release the mouse button. More extensive help can be viewed by selecting the **HTML Help Viewer** command from the **Help** menu.

3.1 Menu bar

The menu bar has **File**, **Edit**, **View**, **Option** and **Help** menus. If you move the pointer over one of the menus, then press and hold down the left mouse button, a pulldown menu appears with further commands. Drag the mouse pointer down over the desired command (look at the context-sensitive help text in the main window title) and release the mouse button to select the command.

3.1.1 File menu

The File menu is used to create new images, to open existing images and to load reflection lists that are displayed. New images that are created can also be saved to disk files. The commands in the file menu are New, Open, Save As, Image Template, Preferences, Print and Exit. Many of these use the image file and template selection dialog shown next.



Figure 2. Image file and template selection dialog.

The image file and template selection dialog is a standard Motif file selection dialog with three additional controls for sequence start, sequence increment and

number of images. In order to specify a range of images, the Filter or Template field is combined with the sequence start and increment fields to yield complete filenames. Any question mark characters (?) are replaced with the sequence number to get a filename. For example, if the Template is image.??? the Seq start is 1 and the Seq incr is 1, then the first image filename is image.001, the next is image.002, and so on.

Here are specific directions on each of the commands in the **File** menu:

New

The New command pops up a menu with the following choices:

- Image File...
- Scan...
- Overlay...
- Average...
- Tile...
- Statistics...

Image file...

When this is activated, the image file selection dialog appears. Select an image file from the list of files and then select **OK**. The file will be read in, automatically scaled to the color table and displayed.

Scan

This command is not yet implemented, therefore it is insensitive and cannot be selected.

Overlay

Overlay creates a new image by overlaying a series of images and displays the resulting image. The image can be saved to disk by selecting the **File Save As...** command. To use this command, first read in and display an image with either the **File Open...** or **File New Image File...** commands. Then select **File New Overlay...** The image file and template selection dialog appears (see Figure 2). Enter the sequence number start, the sequence number increment, the number of images to overlay and the image filename template, then select **OK, overlay**. The image filenames will be constructed from the template and a sequence number by replacing the question marks (?) in the template with the sequence number. For example, an image template ribosome.???? might represent the files ribosome.0001, ribosome.0002, etc.

When images are overlaid, the highest pixel value in any input image for a given position is placed in the created image. For example, if you have three 2x2 images with pixel values:

1	2	2	3	4	3
3	4	4	1	2	1

the overlay result image will be:

4	3
4	4

The overlay command is useful to create pseudo rotation images that span a larger volume of reciprocal space.

If an error occurs while reading in an image, then the overlaying stops and the overlaid image in its current state is displayed. Use **File Save As...** to save the overlay image to disk.

Average

Average is similar to the overlay command and works the same way. Instead of selecting the highest pixel value for the output image, the average value is calculated. In addition, an image of standard deviations is calculated and written to the file named by the first image used in the averaging with the characters 'sd' appended. Use **File Save As...** to save the average image to disk.

Tile

Tile creates a new image by tiling a series of images and displays the resulting image. The image can be saved to disk by selecting the **File Save As...** command. To use this command, first read in and display an image with either the **File Open...** or **File New Image File...** commands. Then select **File New Tile....** The image file and template selection dialog appears (see Figure 2 above). Enter the sequence number start, the sequence number increment, the number of images to overlay and the image filename template, then select **OK, tile**. The image filenames will be constructed from the template and a sequence number by replacing the question marks (?) in the template with the sequence number.

The tiled image has the same dimensions and data type as the input images. In order to achieve this, some pixels in the input images are skipped over. Zooming in on a displayed tiled image does not recover the pixels skipped over. The number of rows and columns in the output image is a function of the number of images. For example, if 4 images are tiled, then the tiled image will be 2x2. If 6 images are tile, then the tiled image will be 2 x 3 with gaps along each row. To avoid gaps, choose a number of images that is the square of an integer (namely,

4, 9, 16, 25, ...). Furthermore, the first image is placed at position (0, 0) while the next image is adjacent to the first image on the same row. Rows run along the *FAST* pixel direction while columns run down the *SLOW* pixel direction. In the display, the *FAST* and *SLOW* directions may have a non-standard orientation (see Orientation in section 3.1.2.1).

If an error occurs while reading in an image, then the tiling stops and the tiled image in its current state is displayed.

Statistics

When implemented, Statistics will statistically analyze an image or a series of images. For now, to average images and compute a pixel by pixel standard deviation, use the **Average** command.

Open...

The **File Open...** command selects an image file to display. When this is activated, the image file selection dialog appears. Select an image file from the list and then select **OK, Open**. The file will be read in, automatically scaled to the color table and displayed.

Load reflnlist

The **File Load reflnlist...** command selects a reflection list to display. When this is activated, a file selection dialog box appears. Select a reflection list file from the list and then select **OK**. The file will be read in and displayed. If reflections are not displayed, check that **Observed reflns** is set in the **View** menu. Check also that the reflection list contains the fields `fObs_pixel0` and `fObs_pixel1`, or `fCalc_pixel0` and `fCalc_pixel1`.

The properties of the reflection list display can be edited by selecting **Edit reflection properties...** in the **Edit** menu (see 3.1.2).

Save As...

This command saves the currently displayed image to a file. If there is no currently displayed image, then this command does nothing. The image file selection dialog appears which allows you to enter a filename for the new image. If you select a filename that already exists, the new image will overwrite the file unless the old file is write protected.

Image template...

This command lets you view and edit the current settings for the image filename template and sequence numbers. An image template dialog box appears as shown in Figure 2 above. The settings in the dialog box come from

the header of the last image read in or from the last time the image template dialog appeared. Remember that **Overlay**, **Average**, **Tile** and **Statistics** in the **File New** menu all use the image file and template selection dialog.

Exit

This command exits from dtdisplay.

3.1.2 Edit menu

The edit menu has commands to display dialog boxes that let you edit the way images and reflections are displayed.

3.1.2.1 Image view properties dialog

Choose **Image View Props...** in the **Edit** menu to display a dialog box with the current image view properties as shown in Figure 3. The view properties include Integration box size, Saturated pixel cutoff, Saturated pixel color, White pixel in image, Black pixel in image, Image border fraction, Orientation, Preserve aspect ratio, and Pixel aspect ratio. In all cases the values set in this dialog do not take effect until either the **OK** or **Apply** button is selected. Also, if no image is currently displayed, then settings made in this dialog box have no effect. Selecting **Help** activates the HTML viewer for on-line help.

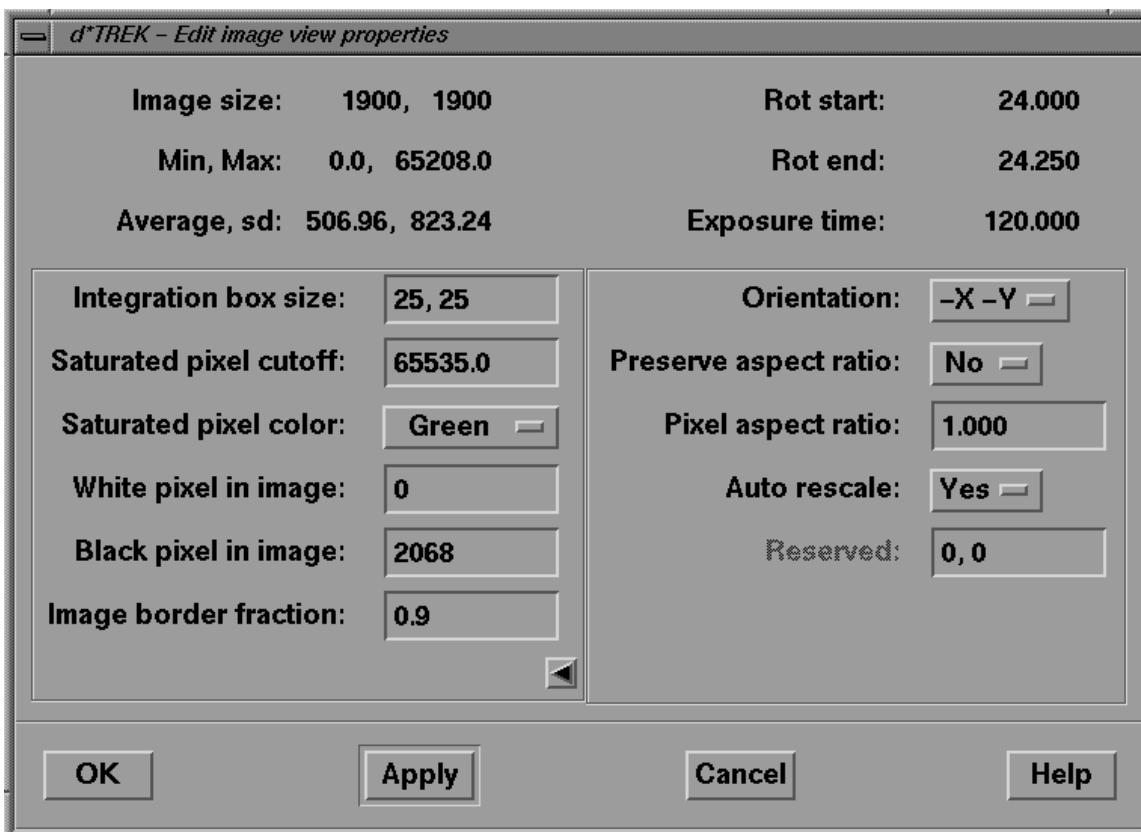


Figure 3. Image view properties dialog.

The dialog box has two halves -- a left one and a right one. To view the full dialog box, select the small arrow button below the Image border fraction field. To return to view to the left half, select the arrow button again. The top portion of the dialog shows static image properties calculated from the pixel values in the image itself. The lower portion shows the properties that you can change which will affect the displayed image.

STATIC IMAGE PROPERTIES

Image size

The number of pixels in the fast and slow directions in the image.

Min, max

These are the minimum and maximum pixel values in the image excluding a border of 10% around all 4 sides.

Average, sd

These are the pixel value average and standard deviation calculated from the image excluding a border of 10% around all 4 sides.

Rot start

This is the relative rotation angle start from the image header.

Rot end

This is the relative rotation angle end from the image header

Exposure time

This is the image exposure time in seconds from the image header.

ADJUSTABLE IMAGE PROPERTIES

Integration Box Size

Enter the integrate box size in pixels in the *FAST* and *SLOW* image directions separated by a comma (,). When a pixel is selected in the image display area, then the pixels in an area I_x by I_y are integrated. First an average and standard deviation is computed, then after excluding pixels with a value 3sigma above that average a new average and standard deviation is computed. Then all the pixels in the area are summed and the average pixel value times the number of pixels is subtracted. This gives a rough estimate of the integrated intensity of a picked Bragg reflection. If the non-uniformity information is available, then the area is corrected for non-uniformity first. If the Integrate Box Size is less than 2 x 2, then a default value of 24 x 24 is used. No effort is made to exclude neighboring spots from the integrate box when the calculations are made.

This box size is also used to set the size of the small zoomed image in the lower left of the main **dtdisplay** window. The box size is local only to dtdisplay. It does not affect the box size in **dtintegrate** or **dtfind**.

Saturated Pixel Value

Enter the minimum value of a saturated pixel. All pixel values that are greater than this value will be displayed in the Saturated Pixel Color.

Saturated Pixel Color

Select the color used to display saturated pixels from the combo box. A limited number of colors is available.

White pixel in image (Minimum scale value)

Pixel values below this threshold are all set to white. This value may also be changed through the color scale below the image.

Black pixel in image (Maximum scale value)

Pixel values above this threshold are all set to black. This value may also be changed through the color scale below the image.

NOTE on color scale mapping:

If the **White pixel** value is more than the **Black pixel** value, then the color scaling is reversed. Thus, if these values are 10 and 5000, then the color mapping runs from white (low) to black (high). If these values are 5000 and 10, then the color mapping runs from white (high) to black (low) which reverses the color mapping.

The starting values for the white and black thresholds are set automatically whenever a image is read in with the **Open** command in the **File** menu. The average and standard deviation of all pixels except a border of 10% along the four edges is calculated. The white threshold is set to the average - 1.5 * standard deviation. The black threshold is set to the average + 5.0 * standard deviation.

The color scale is most easily changed by directly manipulating the minimum and maximum scale values on a color scale which appears below the large image display area. If the Image border fraction is less than 1, then a color scale control pops-up when the left mouse button is pressed when the cursor is in the line profile area along the bottom of the image area. You can change the minimum scale value or the maximum scale value by grabbing the end of the arrow and moving it left or right. Values for the displayed image are updated only when the left mouse button is released in the bottom line profile area. To cancel any impending changes, move the cursor outside of the bottom line profile area and release the left mouse button. The color scale has marks for 3 values: the image average is a solid vertical line, 1.5 sigma below the average is an ascending half line to the left of the average (it may be off scale), 5 sigma above the average is a descending half line to the right of the average (it may be off scale).

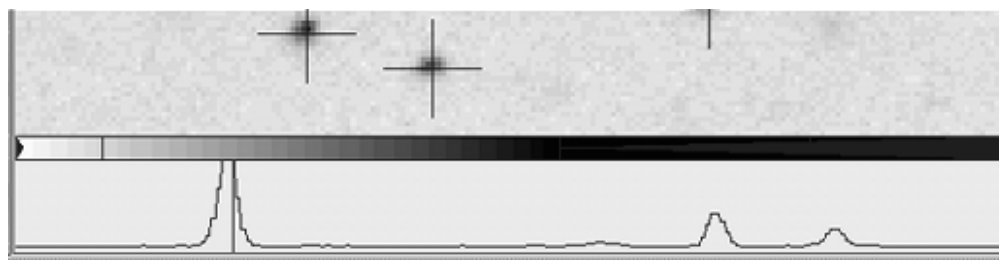


Figure 4. Example of color scale control.

Image border fraction

If this is less than 1, a border is reserved on the right and bottom sides of the image display area for drawing line profiles. A value of 0.9 results in the image area occupying 90% of the drawing area with the line profiles occupying strips of 10% on the right and bottom sides.

Resource file default: *cUlimageviewprops*tfBorder.value: 0.9

Orientation

A combo box is used to select one of eight possible orientations for the image display. The choices show how the image FAST and SLOW pixel directions should be mapped to the X Window coordinate system X and Y. X runs across the display while Y runs down the display. Do not confuse the X Window display XY coordinates with the laboratory XY coordinates. The selected orientation is applied only after selecting the OK or Apply button.

Resource file default: *cUlimageviewprops*omOrient*labelString: +X +Y

Preserve aspect ratio

Select whether the Pixel Aspect Ratio will be preserved when displaying images. This is useful when you want to preserve the image aspect ratio when you resize the image display area.

Pixel aspect ratio

Enter the pixel size aspect ratio. The ratio is the size of the pixel in the FAST direction divided by the size of the pixel in the SLOW direction: FAST / SLOW. For example, if the detector pixels are 1.28 mm by 2.00 mm, then enter 0.64 for the aspect ratio.

Auto rescale

This is reserved for future use and has no effect now.

Reserved

This is reserved for future use and has no effect now.

3.1.2.2 Reflection view properties dialog

Choose **RefIn View Props...** in the **Edit** menu to display a dialog box with the current reflection view properties as shown in Figure 4. ***This dialog box is not active yet.*** The view properties include Symbol, Size, Color, Visibility, Restrict, and Rotation range. In most cases the values set in this dialog do not take effect until either the **OK** or **Apply** button is selected. Also, if no image is currently displayed, then settings made in this dialog box have no effect. Selecting **Help** activates the HTML viewer for on-line help.

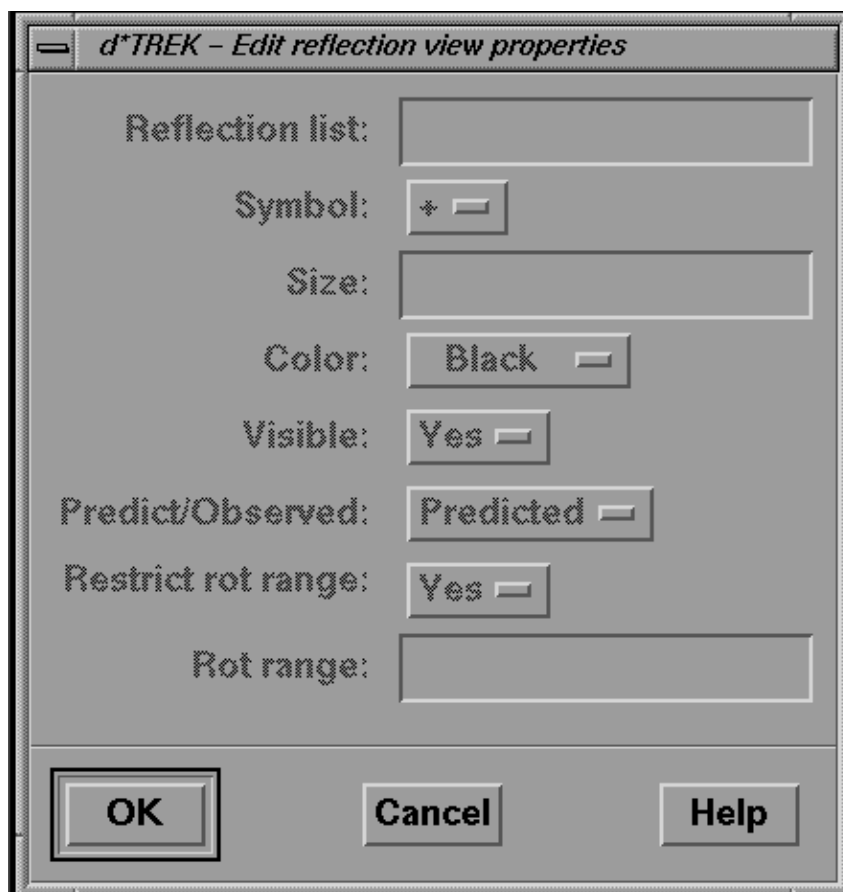


Figure 5. Reflection view properties dialog.

Reflection list

This shows the file name of the reflection list. Use **File Load refln list...** to load the reflection list.

Symbol

Select the symbol used to draw reflections with. You can choose a plus sign, an X (cross), a circle or a box. **Only a plus sign is supported now!**

Size

Enter the size of the symbol in pixels. **The size is fixed at 20!**

Color

Select the color of the symbol. The default color is set by the reflnPlotColor resource (see section 5.2).

Visibility

Select whether this reflection list will be plotted (visible) or not.

Restrict rotation range

Select whether all reflections in a list should be plotted or whether only reflections that fall with a restricted rotation range.

Rotation range

Enter if the restricted range for reflections to be plotted. This range is only used if Restrict rotation range is in effect.

Clear selected output

When this command is activated, any selected text in the scrolled text output window is deleted. Text is selected in the normal Motif style: double-, triple- or quadruple-clicking the mouse button, dragging the mouse pointer while holding down the shift key, etc.

3.1.3 View menu

Commands in the **View** menu are quick ways to change what is viewed in the image display area.

Refresh

This refreshes the image display area by remapping from image pixel values to the color scale and displaying the result.

Zoom...

This does no zooming, but pops up a message box describing how to zoom by dragging out a zoom box with the middle mouse button. The message

box serves as a reminder of the not-so-apparent function of the middle mouse button. **NOTE:** Make sure the CAPS LOCK, SHIFT and CTRL keys are not depressed or you will be unable to drag out a zoom box.

Unzoom

This unzooms a zoomed image and refreshes an unzoomed image. You can also unzoom an image by double-clicking on it with the middle mouse button.

Pan...

This does no panning, but pops up a message box describing how to pan by dragging the cursor with both the left AND middle mouse button pressed.. The message box serves as a reminder of the not-so-apparent function of the mouse buttons.

Rescale

This is not implemented. This should recalculate a default color scale by computing the average and standard deviation of pixels in the image and scaling them to a gray scale. See section 3.1.2.1 **Note on color mapping** for more information.

Image header...

This displays a new window with the image header in it. You cannot save any edits made to the image header.

Calculated reflections

Observed reflections

These toggle buttons select whether any available reflection list is displayed or not. Only Observed works for now.

Resolution arcs

This draws up to 5 iso or constant resolution arcs on the main displayed image. To determine the resolution of any given arc, simply click on it and view the resolution in the scrolled text output in the upper left of the main window.

Measure

This selects whether the crystallographic d-spacing in Ångstroms between the last two selected pixels is calculated and displayed in the scrolled text output area. The detector position and spatial distortion as well as the source wavelength are required to calculate the d-spacing. If they are unavailable, then the d-spacing is not calculated. **Shortcut:** You may also use Shift-Middle mouse button to measure the d-spacing between any two points.

3.1.4 Option menu

The **Option** menu provides a way to easily add extensions to the dtdisplay program. Presently, no extensions are implemented.

Perspective

This creates a separate window with a perspective or mesh plot of the currently displayed image area. Not yet implemented.

Contour

This creates a separate window with a contour plot of the currently displayed image area. Not yet implemented.

3.1.5 Help menu

This help menu provides access to the hyper-text markup language viewer. Context-sensitive help is provided in the main window title whenever a button is armed. (In Motif version 1.2, the title will also provided help on text fields when they gain keyboard focus.)

Help via the HTML viewer

Select this command to activate a hyper-text markup language (HTML) viewer to view the dtdisplay HTML help files. The default HTML viewer is NCSA Mosaic, but this may be changed in the resource file. If the viewer is already running in the process tree, then it is used, otherwise a new process is created that runs the viewer.

3.2 Main window controls

Along the left side of the main window below the menu bar is a control panel. The panel contains a scrolled text output area, text fields to change the detector position and wavelength and buttons to move to the next or previous image in a scan.

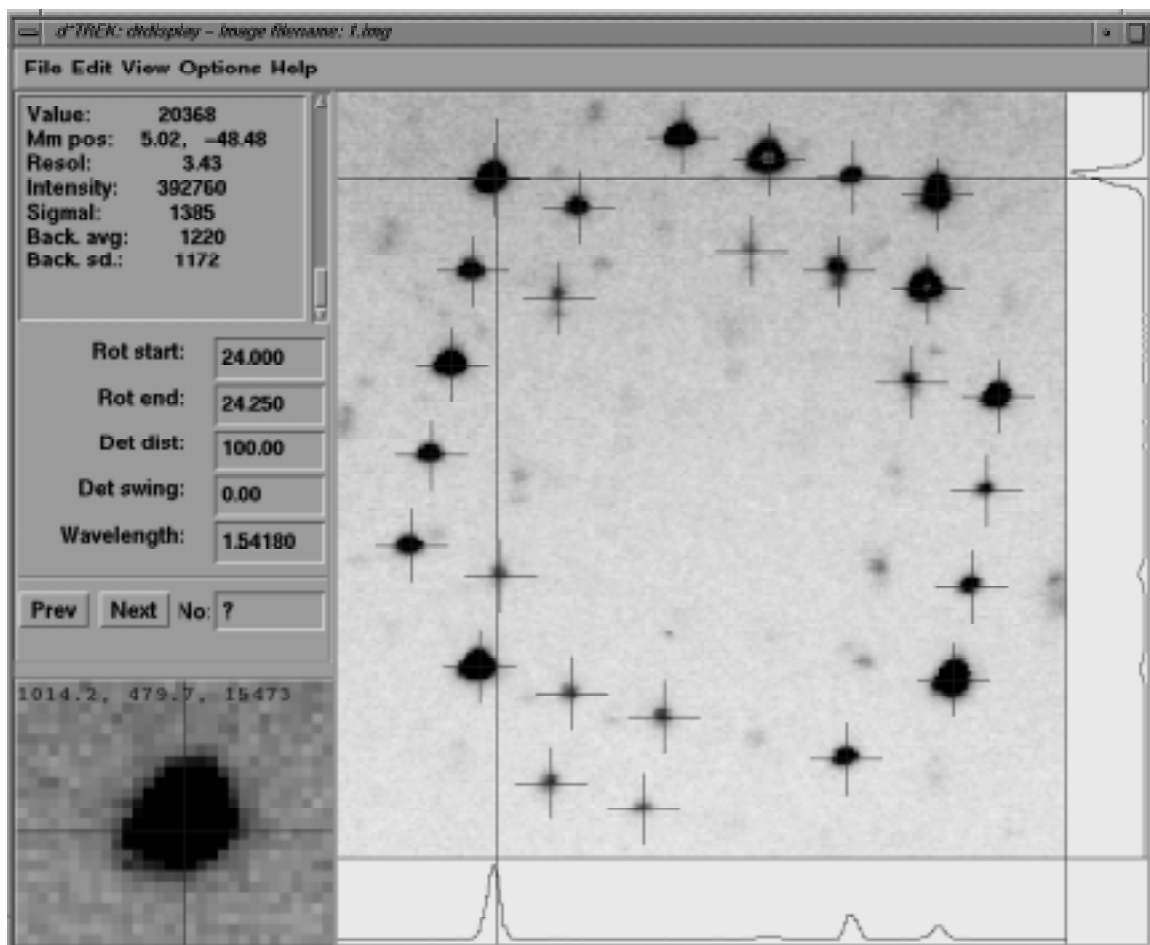


Figure 6. dtdisplay main window. Note scrolled text area and other controls on left.

Scrolled Text Output

The scrolled text output is a flexible way to provide information about points selected in the large image display area. When you select pixels in the large image area, the information in the scrolled text area usually contains:

- Pixel position
- Pixel value
- Millimeter position

Resolution
Integrated intensity and standard deviation
Average background and standard deviation

If the **Measure** toggle button in the **View** menu is set, then the crystallographic d-spacing between the selected pixel and the previously selected pixel will be reported. The information reported does depend on a number of factors. The millimeter position, resolution and d-spacing all depend on knowledge of the detector position, the detector spatial distortion and the source wavelength. If this additional information is unavailable, then millimeter position, resolution and d-spacing will not be reported.

Shortcut: You may also use Shift-Middle mouse button to measure the d-spacing between any two points.

Use the scroll bar to scroll the output backwards and forwards. The output may be cleared by first selecting it and then choosing the **Clear selected output** command from the **Edit** menu.

Rot. start, Rot. end

Enter the rotation start angle and end angle for the image. Presently, these have no effect.

Det swing

Enter the desired detector swing angle in degrees. The direction of the swing axis is specified in the header and is usually the theta or 2theta value of the detector. Use this command to check the resolution limits at a particular detector position. The starting value for the swing angle is read from the image header. The value is updated *only after a carriage return or when the text field loses input focus*. See **Note** under **Wavelength** below.

Det dist

Enter the desired detector to sample distance in millimeters. Use this command to check the resolution limits at a particular detector position. The starting value for the distance is read from the image header. The value is updated *only after a carriage return or when the text field loses the input focus*. See **Note** under **Wavelength** below.

Wavelength

Enter the source wavelength in Ångstroms. This affects the calculation of resolution only.

Note: When the detector distance or swing angle or the source wavelength is changed, then the minimum and maximum resolution available on the detector is displayed in the scrolled text output in the upper left of the main window.

Prev, Next, No.

These commands let you easily select the previous image, the next image or a specific image sequence number in a scan to read in and view. If a scan definition is not found in the image header, then these commands should be insensitive. You may specify your own scan definition with the **Image template...** command in the **File** menu. When an image is read with the **Prev**, **Next** or **No.** commands, the mapping of pixel values to colors is not changed, but the current mapping is used. If an error occurs when reading the image file, then the image display areas go blank. **No.** uses the specific sequence number, no matter what the sequence increment is. **Prev** and **Next** subtract or add the sequence increment to the current sequence number to get the sequence number of the image to display.

3.3 Image display areas

dtdisplay has two image display areas, a large one on the right and a small one in the lower left of the main window (see Figure 6 above). A few commands work in both display areas, while other commands only work in the large display area.

Left mouse button pressed down

If you press and hold the left mouse button down in an image display area, the image pixel coordinates and pixel value are displayed in the upper left or upper right of the display area. As you move the mouse pointer with the left mouse button held down, the information is updated. This is the fastest way to see pixel values. This works in both the large and small image display areas.

Left mouse button released

When you release the left mouse button in the large image display area, this selects the pixel under the mouse pointer. Information about the pixel is displayed in the scrolled text output area and usually includes pixel coordinate, pixel value, resolution, etc. (see above). Furthermore, the small image display area shows a zoomed region around the selected pixel.

Middle mouse button

Use the middle mouse button to select a rectangle to zoom in on. When you press and hold down the middle mouse button a rectangle appears that indicates the area to be zoomed. Drag the mouse pointer to change the size of the rectangle. Releasing the middle mouse button displays the zoomed area. Double-clicking the middle mouse unzooms the image or refreshes an unzoomed image. If the zoom box is within a single pixel, this also unzooms the image. If you zoom large enough, the actual pixel values are displayed within the pixels. The middle mouse button works in both the large and small image display areas.

If you press and hold the left mouse button, while the zoom rectangle is active, then this pans or moves the zoom rectangle around within the image. You may release only the left mouse button to continue to change the size of zoom box.

NOTE: Make sure the CAPS LOCK, SHIFT and CTRL keys are not depressed or you will be unable to drag out a zoom box.

FIRST left, then middle mouse button pressed down

If you press and hold the left mouse button down in an image display area, then press and hold the middle mouse button down, you can pan around a zoomed image by dragging the mouse around. This works best in the small image area.

Right mouse button

The right mouse button currently has no function. Perhaps in the future it will pop-up a menu.

Keyboard commands in the image areas

If the keyboard focus is in the large or small image area, then a few keyboard commands can be used. One way to set the keyboard focus in the small image area is to click on the Scan No. text field, then press the Tab key. To set the keyboard focus to the large image area, select the scrolled text area, then press Ctrl+Tab. Keyboard commands are:

>, G, <, L

These zoom (>, G) or unzoom (<, L) the displayed image by a factor of 2.

I, J, K, M. arrow keys

These shift or pan the view. The shift is one-tenth of the viewed area or 1 pixel, whichever is larger.

Figure 8 was made by the following sequence of commands:

- File Open...
- Drag out a zoom box with middle mouse button in the large image area
- Drag out a zoom box again on the zoomed image to zoom even more.
- Click on the peak with the left mouse button so the small image area displays it.

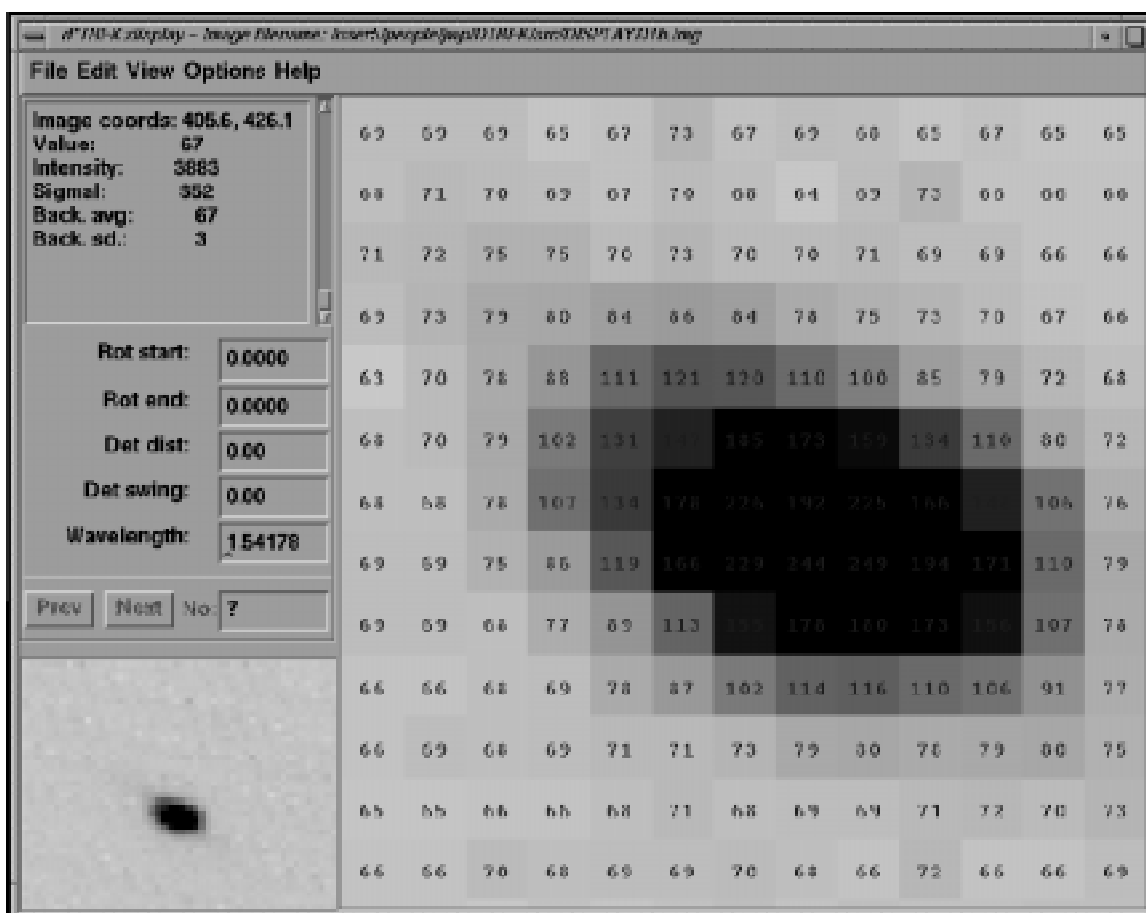


Figure 8. Large area zoomed view with pixel values displayed. Small area zoomed view.

4. Overcoming difficulties

Difficulties with dtdisplay can be divided into problems with X Windows and problems with dtdisplay itself.

4.1 Problems with X Windows

If you cannot get the main window to appear at all, chances are the DISPLAY environment variable or the permissions on the display are not set properly. Consult you system manager for help on this.

If the main window appears, but the message “not enough colors” appears, then you have a black and white display on which dtdisplay will not work. Move to a color display and try again.

If the main window appears, but the image colors look funny and you experience “color flashing”, then there is a problem with your X server. dtdisplay contains algorithms to reduce color flashing, but some X servers or window managers (e.g. Xsgi) do not install colormaps according to the X and Motif standards. To work around this, you can exit from other programs that use a large number of colors and restart dtdisplay.

If the main window appears, but all the buttons have funny names like cbFile, then the application defaults file was not loaded properly. Be sure the application defaults file Dtdisplay is in the proper directory such as /usr/lib/X11/app-defaults or your home directory. Or use xrdb -merge Dtdisplay to explicit load the file, then restart dtdisplay.

4.2 Problems with dtdisplay

Since dtdisplay gets almost all of its input information from an image header, difficulties may arise when the header is incomplete. Be sure that the detector spatial distortion information and non-uniformity information is correct and the files exist in the current directory or the filenames in the header include the path or directory.

The output of dtdisplay clearly shows when any files have not been read properly. Select **View Image header...** command to display image header with its source, goniometers, detector and rotation properties. If these are incorrect, then the image header is incorrect and should be modified. In these cases, add or correct the information (use dtheadredit) and run dtdisplay again.

Many commands in dtdisplay have no effect if an image is not currently displayed. If the image display areas are gray, then read in an image with the **Open...** command in the **File** menu. If the scan information (**File, Image Template...**) points to images of a different size than the currently displayed image, then any new images read with any command that uses the scan

information will be displayed in the size of the current image. To prevent this problem, always read in at least one image from the scan with the **File Open...** command before using the **File New Overlay** and **File New Tile** commands.

If you cannot drag out a zoom box with the middle mouse button, be sure the CAPS LOCK or SHIFT buttons are not depressed.

5 Installation

dtdisplay has been written in C++ with the ICS BuilderXcessory program on a Silicon Graphics Indigo running IRIX 4.0.5H or higher and Motif version 1.1 or higher. We used the GNU g++ compiler and the GNU make utility. Source code, makefiles, the application defaults file and the html files are necessary to build dtdisplay. In order to provide a very high level of customization, dtdisplay extensively uses X resources. For this reason the application defaults file is very important and must be installed properly. The program uses class Dtdisplay, and the application defaults file is also called Dtdisplay. This file should be placed in your home directory or in the system-wide application defaults directory `/usr/lib/X11/app-defaults`

To install dtdisplay from the sources, see 5.2 Make.

If you have the dtdisplay executable binary, then place it in a standard location in your path (`/usr/local/bin` comes to mind).

Place the html files in a common area (`/usr/local/html` comes to mind) where all users have read permission.

Edit the application defaults file to reflect your customization (see 5.1 Application defaults) and place it in your login directory or in the system application defaults directory.

5.1 Application defaults file

All the titles, labels, fonts, error messages, help messages and colors used by dtdisplay are specified in an application defaults file. This allows every site to customize the appearance of dtdisplay. With a little bit of work, but without re-compiling dtdisplay, the entire user interface can be changed to French or any other language. Despite the apparent advantage of extensive customization, a major disadvantage is that the application defaults file must exist and be

accessible to all who use dtdisplay. This section describes the defaults file and helps you customize it.

dtdisplay defines the class Dtdisplay and hence the application defaults file is called Dtdisplay. Below is the application defaults file as distributed.

```
1  !/*
2  ! * RCS stuff:
3  ! * $Author: jwp $
4  ! * $Date: 96/04/01 11:27:45 $
5  ! * $Header: /u6/jwp/DTREKREPOSITORY/DTREK/DTREK/src/DTDISPLAY/Dtdisplay,v 1.4 96/04/01 11:27:45 jwp
   Exp $
6  ! * $Log:   Dtdisplay,v $
7  !Revision 1.4 96/04/01 11:27:45 jwp
8  !Changes due to differences between servers: Motif 1.1 vs 1.2?
9  !
10 !Revision 1.2 95/11/08 17:38:48 jwp
11 !Change pushButton4.labelString.
12 !
13 # Revision 1.1 95/09/07 10:34:07 jwp
14 # Initial commit.
15 #
16 ! * $Revision: 1.4 $
17 !*/
18 !
19 ! Generated by the ICS Builder Xcessory (BX).
20 !
21 ! Version = Builder Xcessory 3.5.1
22 !
23 *fontList:      -*-helvetica-bold-r-*-18-*-
24 *XmLabel.fontList:  -*-helvetica-bold-r-*-18-*-
25 *dalmageBig.resizable:  True
26 *dalmageSmall.resizable:  True
27 *frame.resizable:  True
28 *frame.leftPosition:  52
29 *dalmageBig*font:  -*-helvetica-bold-r-*-18-*-
30 *dalmageBig*foreground:  blue
31 *dalmageBig*Foreground:  blue
32 *dalmageBig.translations:  #override\n\
33 <Btn1Motion>:  DrawingAreaInput() ManagerGadgetButtonMotion() \n\
34 <Btn2Motion>:  DrawingAreaInput() ManagerGadgetButtonMotion() \n\
35 <BtnUp>:  DrawingAreaInput() ManagerGadgetButtonMotion() \n\
36 <BtnDown>:  DrawingAreaInput() ManagerGadgetButtonMotion()
37 *dalmageSmall.translations:  #override\n\
38 <Btn1Motion>:  DrawingAreaInput() ManagerGadgetButtonMotion() \n\
39 <Btn2Motion>:  DrawingAreaInput() ManagerGadgetButtonMotion() \n\
40 <BtnUp>:  DrawingAreaInput() ManagerGadgetButtonMotion() \n\
41 <BtnDown>:  DrawingAreaInput() ManagerGadgetButtonMotion()
42 *keyboardFocusPolicy:  explicit
43 *dalmageBig*traversalOn:  True
44
45 *helpLineHTMLHelp:  Use NCSA mosaic to view html help files
46 ! Mosaic Help default overrides:
47 *helpDirectory:  /u6/jwp/DTREK/src/DTDISPLAY/
48 *htmlFileOpen:  FileOpen
49 *htmlHelpHTML:  dtdisplay
50 *helpDelimiter:  \ -- Help: \
```

```
50 *helpPrefix:
51 *pixelValueColor:    blue
52 *reflnPlotColor:    blue
53 !
54 !
55 !
56 *dtdisplay.title: d*TREK - dtdisplay
57 *dtdisplay.iconName: dtdisplay
58 *dtdisplay*mbError.dialogTitle: d*TREK - Error dialog
59 *dtdisplay*mbError.messageString: This is an error message
60 *dtdisplay*mbError.cancelLabelString: Dismiss
61 *dtdisplay*mbProgress.dialogTitle: d*TREK display - Progress ...
62 *dtdisplay*mbProgress.messageString: Progress of overlaying images
63 *dtdisplay*fsbLoadRefln.dialogTitle: d*Trek - display: File: Load reflection file
64 *dtdisplay*fmViewHeader.dialogTitle: d*TREK - Image header
65 *dtdisplay*pbViewHeaderDismiss.labelString: Dismiss
66 *dtdisplay*tfViewHeader.scrollLeftSide: True
67 *dtdisplay*xmDialogShell3.title: d*TREK - How to Zoom
68 *dtdisplay*mbViewZoomPan.dialogTitle: d*TREK - How to Zoom and Pan ...
69 *dtdisplay*mbViewZoomPan.messageString: :::"Select the image area to ZOOM" by dragging out a box in the
image with the middle mouse button. To UNZOOM an image, single click on it with the middle mouse
button. To PAN an image, press and hold the left mouse, then press and hold the middle mouse button;
drag the mouse to pan."
70 *dtdisplay*pbFileNewImage.labelString: Image file...
71 *dtdisplay*pbFileNewScan.labelString: Scan...
72 *dtdisplay*pbFileNewOverlay.labelString: Overlay...
73 *dtdisplay*pbFileNewTile.labelString: Tile...
74 *dtdisplay*pbFileNewUnderlay.labelString: Underlay...
75 *dtdisplay*pbFileNewAverage.labelString: Average...
76 *dtdisplay*pbFileNewStats.labelString: Statistics...
77 *dtdisplay*cbFileNew.labelString: New
78 *dtdisplay*pbFileOpen.labelString: Open...
79 *dtdisplay*pbFileLoad.labelString: Load refln list...
80 *dtdisplay*pbFileSaveAs.labelString: Save As...
81 *dtdisplay*pbFileTemplate.labelString: Image template...
82 *dtdisplay*pbFilePrefer.labelString: Preferences...
83 *dtdisplay*pbFilePrint.labelString: Print...
84 *dtdisplay*tbFileListen.labelString: Respond to updates
85 *dtdisplay*pbFileExit.labelString: Exit
86 *dtdisplay*cbFile.labelString: File
87 *dtdisplay*pbEditImage.labelString: Image view props...
88 *dtdisplay*pbEditRefln.labelString: Refln view props...
89 *dtdisplay*pbEditClear.labelString: Clear selected output
90 *dtdisplay*cbEdit.labelString: Edit
91 *dtdisplay*pbViewRefresh.labelString: Refresh
92 *dtdisplay*pbViewZoom.labelString: Zoom...
93 *dtdisplay*pbViewUnzoom.labelString: Unzoom
94 *dtdisplay*pbViewPan.labelString: Pan...
95 *dtdisplay*pbViewHeader.labelString: Image header...
96 *dtdisplay*tbViewCalc.labelString: Calculated reflns
97 *dtdisplay*tbViewObs.labelString: Observed reflns
98 *dtdisplay*tbViewReso.labelString: Resolution arcs
99 *dtdisplay*tbViewMeasure.labelString: Measure
100 *dtdisplay*tbViewPlot.labelString: Plot points...
101 *dtdisplay*tbViewPixel.labelString: View pixel values...
102 *dtdisplay*cbView.labelString: View
103 *dtdisplay*pbOptionsPerspect.labelString: Perspective...
104 *dtdisplay*pbOptionsContour.labelString: Contour...
105 *dtdisplay*cbOptions.labelString: Options
106 *dtdisplay*pbHelpHTML.labelString: Help via HTML viewer...
```

```
107 *dtdisplay*cbHelp.labelString: Help
108 *dtdisplay*damageSmall.background: grey95
109 *dtdisplay*lbImageNum.labelString: No:
110 *dtdisplay*pbImageNext.labelString: Next
111 *dtdisplay*pbImagePrev.labelString: Prev
112 *dtdisplay*lbWavelength.labelString: Wavelength:
113 *dtdisplay*tfOutput.value: This scrolled text area\n
114 contains general information\n\n
115 on selected user actions.
116 *dtdisplay*lbRotStart.labelString: Rot start:
117 *dtdisplay*lbRotEnd.labelString: Rot end:
118 *dtdisplay*lbDetDist.labelString: Det dist:
119 *dtdisplay*lbDetSwing.labelString: Det swing:
120 *dtdisplay*damageBig.background: grey95

121 !+++++
122 ! Builder Xcessory class defaults section
123 !+++++

124 !++
125 ! Defaults for class: CUIPSfileSelect
126 !--
127 CUIPSfileSelect.CUIPSfileSelect.dialogTitle: Select PostScript file name
128 CUIPSfileSelect.CUIPSfileSelect.selectionLabelString: PostScript file
129 CUIPSfileSelect.CUIPSfileSelect.okLabelString: OK
130 CUIPSfileSelect.CUIPSfileSelect.filterLabelString: Filter
131 CUIPSfileSelect.lbComment.labelString: Comment

132 !++
133 ! Defaults for class: CUNumberField
134 !--
135 CUNumberField.CUNumberField.sensitive: True
136 CUNumberField.scrollBar.value: 1
137 CUNumberField.scrollBar.minimum: -500
138 CUNumberField.scrollBar.maximum: 611
139 CUNumberField.scrollBar.pageIncrement: 10
140 CUNumberField.textField.value: %.2f
141 CUNumberField.label.labelString: label

142 !++
143 ! Defaults for class: CUlom8
144 !--
145 CUlom8.pb0.sensitive: True
146 CUlom8.pb0.labelString: Black
147 CUlom8.pb1.sensitive: True
148 CUlom8.pb1.labelString: Red
149 CUlom8.pb2.sensitive: True
150 CUlom8.pb2.labelString: Orange
151 CUlom8.pb3.sensitive: True
152 CUlom8.pb3.labelString: Yellow
153 CUlom8.pb4.sensitive: True
154 CUlom8.pb4.labelString: Green
155 CUlom8.pb5.sensitive: True
156 CUlom8.pb5.labelString: Blue
157 CUlom8.pb6.sensitive: True
158 CUlom8.pb6.labelString: Magenta
159 CUlom8.pb7.sensitive: True
160 CUlom8.pb7.labelString: Cyan
161 CUlom8.pb8.sensitive: True
162 CUlom8.pb8.labelString: White
```

```
163    CUlom8.om9.sensitive: True
164    CUlom8.om9.labelString: Red

165    !++
166    ! Defaults for class: CUlom2
167    !--
168    CUlom2.pb0.sensitive: True
169    CUlom2.pb0.labelString: Black
170    CUlom2.pb1.sensitive: True
171    CUlom2.pb1.labelString: Red
172    CUlom2.pb2.sensitive: True
173    CUlom2.pb2.labelString: Orange
174    CUlom2.pb3.sensitive: True
175    CUlom2.pb3.labelString: Yellow
176    CUlom2.pb4.sensitive: True
177    CUlom2.pb4.labelString: Green
178    CUlom2.pb5.sensitive: True
179    CUlom2.pb5.labelString: Blue
180    CUlom2.pb6.sensitive: True
181    CUlom2.pb6.labelString: Magenta
182    CUlom2.pb7.sensitive: True
183    CUlom2.pb7.labelString: Cyan
184    CUlom2.pb8.sensitive: True
185    CUlom2.pb8.labelString: White
186    CUlom2.om9.sensitive: True
187    CUlom2.om9.labelString: Blue

188    !++
189    ! Defaults for class: CUlom9
190    !--
191    CUlom9.pb0.sensitive: True
192    CUlom9.pb0.labelString: Black
193    CUlom9.pb1.sensitive: True
194    CUlom9.pb1.labelString: Red
195    CUlom9.pb2.sensitive: True
196    CUlom9.pb2.labelString: Orange
197    CUlom9.pb3.sensitive: True
198    CUlom9.pb3.labelString: Yellow
199    CUlom9.pb4.sensitive: True
200    CUlom9.pb4.labelString: Green
201    CUlom9.pb5.sensitive: True
202    CUlom9.pb5.labelString: Blue
203    CUlom9.pb6.sensitive: True
204    CUlom9.pb6.labelString: Magenta
205    CUlom9.pb7.sensitive: True
206    CUlom9.pb7.labelString: Cyan
207    CUlom9.pb8.sensitive: True
208    CUlom9.pb8.labelString: White
209    CUlom9.om9.sensitive: True
210    !!!!!
211    ! For some reason the class defaults are used, so be careful below!
212    !!!!!
213    CUlom9.om9.labelString: Red

214    !++
215    ! Defaults for class: CUlimageviewprops
216    !--
217    CUlimageviewprops.CUlimageviewprops.dialogTitle: d*TREK - Edit image view properties
218    CUlimageviewprops.CUlimageviewprops.selectionLabelString: Command:
219    CUlimageviewprops.lbExpTimeO.labelString: ?
```

```
220 CUImageviewprops.lbRotEndO.labelString: ?
221 CUImageviewprops.lbRotStartO.labelString: ?
222 CUImageviewprops.lbImgSizeO.labelString: ?
223 CUImageviewprops.lbExpTime.labelString: Exposure time:
224 CUImageviewprops.lbRotEnd.labelString: Rot end:
225 CUImageviewprops.lbRotStart.labelString: Rot start:
226 CUImageviewprops.tfReserve.value: Reserved
227 CUImageviewprops.tfAspect.value: 1.0
228 CUImageviewprops.omRescale.pb1.labelString: No
229 CUImageviewprops.omRescale.pb0.labelString: Yes
230 CUImageviewprops.omUseAspect.pb1.labelString: No
231 CUImageviewprops.omUseAspect.pb0.labelString: Yes
232 CUImageviewprops.omOrient.pb7.labelString: -Y -X
233 CUImageviewprops.omOrient.pb6.labelString: -Y +X
234 CUImageviewprops.omOrient.pb5.labelString: +Y -X
235 CUImageviewprops.omOrient.pb4.labelString: +Y +X
236 CUImageviewprops.omOrient.pb3.labelString: -X -Y
237 CUImageviewprops.omOrient.pb2.labelString: -X +Y
238 CUImageviewprops.omOrient.pb1.labelString: +X -Y
239 CUImageviewprops.omOrient.pb0.labelString: +X +Y

240 CUImageviewprops.lbReserve.labelString: Reserved:
241 CUImageviewprops.lbRescale.labelString: Auto rescale:
242 CUImageviewprops.lbAspect.labelString: Pixel aspect ratio:
243 CUImageviewprops.lbUseAspect.labelString: Preserve aspect ratio:
244 CUImageviewprops.lbOrient.labelString: Orientation:
245 CUImageviewprops.tfBorder.value: 0.9
246 CUImageviewprops.lbBorder.labelString: Image border fraction:
247 CUImageviewprops.tfSatValue.value: 65355
248 CUImageviewprops.lbSatColor.labelString: Saturated pixel color:
249 CUImageviewprops.lbSatValue.labelString: Saturated pixel cutoff:
250 CUImageviewprops.tfBoxSize.value: 25, 25
251 CUImageviewprops.lbBoxSize.labelString: Integration box size:
252 CUImageviewprops.lbBlackPix.labelString: Black pixel:
253 CUImageviewprops.lbWhitePix.labelString: White pixel:
254 CUImageviewprops.tfBlackPix.value: 100
255 CUImageviewprops.tfWhitePix.value: 0
256 CUImageviewprops.lbMinMaxO.labelString: ?
257 CUImageviewprops.lbAvgSDO.labelString: ?
258 CUImageviewprops.lbMinMax.labelString: Min, Max:
259 CUImageviewprops.lbAvgSD.labelString: Average, sd:
260 CUImageviewprops.lbImgSize.labelString: Image size:

261 !++
262 ! Defaults for class: CUImagefileSelect
263 !--
264 CUImagefileSelect.CUImagefileSelect.dialogTitle: Select image file and/or template name
265 CUImagefileSelect.CUImagefileSelect.selectionLabelString: Image file
266 CUImagefileSelect.CUImagefileSelect.okLabelString: OK
267 CUImagefileSelect.CUImagefileSelect.filterLabelString: Filter or Template
268 CUImagefileSelect.numImages.sensitive: True
269 CUImagefileSelect.numImages.scrollBar.minimum: 0
270 CUImagefileSelect.numImages.scrollBar.maximum: 111
271 CUImagefileSelect.numImages.textField.value: %.0f
272 CUImagefileSelect.numImages.label.labelString: ::t"Num\"images\"
273 CUImagefileSelect.seqIncr.sensitive: True
274 CUImagefileSelect.seqIncr.textField.value: %.0f
275 CUImagefileSelect.seqIncr.label.labelString: ::t"Seq\"incr\"
276 CUImagefileSelect.seqStart.sensitive: True
277 CUImagefileSelect.seqStart.textField.value: %.0f
```

```
278    CUImagefileSelect.seqStart.label.labelString: :::t"Seq\\"start"
279    !++
280    ! Defaults for class: CUreflnviewprops
281    !--
282    CUreflnviewprops.CUreflnviewprops.dialogTitle: d*TREK - Edit reflection view properties
283    CUreflnviewprops.lbRotRange.labelString: Rot range:
284    CUreflnviewprops.lbRestrictRange.labelString: Restrict range:
285    CUreflnviewprops.lbPredObs.labelString: Pred / Obsv:
286    CUreflnviewprops.lbVisible.labelString: Visible:
287    CUreflnviewprops.lbColor.labelString: Color:
288    CUreflnviewprops.lbSymbol.labelString: Symbol:
289    CUreflnviewprops.pushButton55.labelString: Yes
290    CUreflnviewprops.pushButton56.labelString: No
291    CUreflnviewprops.pushButton54.labelString: Predicted
292    CUreflnviewprops.pushButton57.labelString: Observed
293    CUreflnviewprops.pushButton60.labelString: Both
294    CUreflnviewprops.pushButton53.labelString: Yes
295    CUreflnviewprops.pushButton58.labelString: No
296    CUreflnviewprops.pushButton52.background: black
297    CUreflnviewprops.pushButton52.labelString: Black
298    CUreflnviewprops.pushButton59.background: white
299    CUreflnviewprops.pushButton59.labelString: White
300    CUreflnviewprops.pushButton64.background: red
301    CUreflnviewprops.pushButton64.labelString: Red
302    CUreflnviewprops.pushButton65.background: orange
303    CUreflnviewprops.pushButton65.labelString: Orange
304    CUreflnviewprops.pushButton66.background: yellow
305    CUreflnviewprops.pushButton66.labelString: Yellow
306    CUreflnviewprops.pushButton67.background: green
307    CUreflnviewprops.pushButton67.labelString: Green
308    CUreflnviewprops.pushButton68.background: blue
309    CUreflnviewprops.pushButton68.labelString: Blue
310    CUreflnviewprops.pushButton69.labelString: Cyan
311    CUreflnviewprops.pushButton70.labelString: Magenta
312    CUreflnviewprops.lbSize.labelString: Size:
313    CUreflnviewprops.pushButton21.labelString: + sign
314    CUreflnviewprops.pushButton22.labelString: x sign
315    CUreflnviewprops.pushButton50.labelString: o sign
316    CUreflnviewprops.pushButton51.labelString: [] box
317    CUreflnviewprops.lbReflnName.labelString: Reflection list:

318    !+++++
319    ! Builder Xcessory class defaults section
320    !+++++

321    !++
322    ! Defaults for class: CUIPSfileSelect
323    !--
324    CUIPSfileSelect.CUIPSfileSelect.dialogTitle: d*TREK - Select PostScript file name
325    CUIPSfileSelect.CUIPSfileSelect.selectionLabelString: PostScript file
326    CUIPSfileSelect.CUIPSfileSelect.okLabelString: OK
327    CUIPSfileSelect.CUIPSfileSelect.filterLabelString: Filter
328    CUIPSfileSelect*lbComment.labelString: Comment

329    !++
330    ! Exposed resources for class instances
331    !--

332    *CUImageviewprops.dialogTitle: d*TREK - Edit image view properties
```

```
333 *cUImageviewprops*lbExpTimeO.labelString: ?
334 *cUImageviewprops*lbRotEndO.labelString: ?
335 *cUImageviewprops*lbRotStartO.labelString: ?
336 *cUImageviewprops*lbImgSizeO.labelString: ?
337 *cUImageviewprops*lbExpTime.labelString: Exposure time:
338 *cUImageviewprops*lbRotEnd.labelString: Rot end:
339 *cUImageviewprops*lbRotStart.labelString: Rot start:
340 *cUImageviewprops*tfBlackPix.value: 100
341 *cUImageviewprops*tfWhitePix.value: 0
342 *cUImageviewprops*tfSatValue.value: 65535.0
343 *cUImageviewprops*tfBoxSize.value: 25, 25
344 *cUImageviewprops*tfReserve.value: Reserved
345 *cUImageviewprops*tfAspect.value: 1.0
346 *cUImageviewprops*tfBorder.value: 0.9
347 *cUImageviewprops*omColor*pb8.labelString: White
348 *cUImageviewprops*omColor*pb8.background: white
349 *cUImageviewprops*omColor*pb8.helpLine: Saturated pixels will be WHITE
350 *cUImageviewprops*omColor*pb7.labelString: Cyan
351 *cUImageviewprops*omColor*pb7.background: cyan
352 *cUImageviewprops*omColor*pb7.helpLine: Saturated pixels will be CYAN
353 *cUImageviewprops*omColor*pb6.labelString: Magenta
354 *cUImageviewprops*omColor*pb6.background: magenta
355 *cUImageviewprops*omColor*pb6.helpLine: Saturated pixels will be MAGENTA
356 *cUImageviewprops*omColor*pb5.labelString: Blue
357 *cUImageviewprops*omColor*pb5.background: blue
358 *cUImageviewprops*omColor*pb5.helpLine: Saturated pixels will be BLUE
359 *cUImageviewprops*omColor*pb4.labelString: Green
360 *cUImageviewprops*omColor*pb4.background: green
361 *cUImageviewprops*omColor*pb4.helpLine: Saturated pixels will be GREEN
362 *cUImageviewprops*omColor*pb3.labelString: Yellow
363 *cUImageviewprops*omColor*pb3.background: yellow
364 *cUImageviewprops*omColor*pb3.helpLine: Saturated pixels will be YELLOW
365 *cUImageviewprops*omColor*pb2.labelString: Orange
366 *cUImageviewprops*omColor*pb2.background: orange
367 *cUImageviewprops*omColor*pb2.helpLine: Saturated pixels will be ORANGE
368 *cUImageviewprops*omColor*pb1.labelString: Red
369 *cUImageviewprops*omColor*pb1.background: red
370 *cUImageviewprops*omColor*pb1.helpLine: Saturated pixels will be RED
371 *cUImageviewprops*omColor*pb0.labelString: Black
372 *cUImageviewprops*omColor*pb0.background: black
373 *cUImageviewprops*omColor*pb0.helpLine: Saturated pixels will be BLACK

374 !!!!!
375 ! For some reason the class defaults are used, so be careful below!
376 !!!!!
377 CUIom9.om9.labelString: Red
378 ! Careful here: sometimes works, sometimes does not depending on server!?:
379 *cUImageviewprops*omColor*labelString: Red

380 *cUImageviewprops*omRescale*pb1.labelString: No
381 *cUImageviewprops*omRescale*pb1.helpLine: Not implemented
382 *cUImageviewprops*omRescale*pb0.labelString: Yes
383 *cUImageviewprops*omRescale*pb0.helpLine: Not implemented
384 *cUImageviewprops*omUseAspect*pb1.labelString: No
385 *cUImageviewprops*omUseAspect*pb1.helpLine: Do NOT preserve the pixel aspect ratio
386 *cUImageviewprops*omUseAspect*pb0.labelString: Yes
387 *cUImageviewprops*omUseAspect*pb0.helpLine: Preserve the pixel aspect ratio
388 *cUImageviewprops*omOrient*pb7.labelString: -Y -X
389 *cUImageviewprops*omOrient*pb7.helpLine: Origin in lower right, Y left, X up
```

```
390 *cUImageviewprops*omOrient*pb6.labelString: -Y +X
391 *cUImageviewprops*omOrient*pb6.helpLine: Origin in upper right, Y left, X down
392 *cUImageviewprops*omOrient*pb5.labelString: +Y -X
393 *cUImageviewprops*omOrient*pb5.helpLine: Origin in lower left, Y right, X up
394 *cUImageviewprops*omOrient*pb4.labelString: +Y +X
395 *cUImageviewprops*omOrient*pb4.helpLine: Origin in upper left, Y right, X down
396 *cUImageviewprops*omOrient*pb3.labelString: -X -Y
397 *cUImageviewprops*omOrient*pb3.helpLine: Origin in lower right, X left, Y up
398 *cUImageviewprops*omOrient*pb2.labelString: -X +Y
399 *cUImageviewprops*omOrient*pb2.helpLine: Origin in upper right, X left, Y down
400 *cUImageviewprops*omOrient*pb1.labelString: +X -Y
401 *cUImageviewprops*omOrient*pb1.helpLine: Origin in lower left, X right, Y up
402 *cUImageviewprops*omOrient*pb0.labelString: +X +Y
403 *cUImageviewprops*omOrient*pb0.helpLine: Origin in upper left, X right, Y down

404 !!!!!
405 ! For some reason the class defaults are used, so be careful below!
406 !!!!!
407 CUlom8.om9.labelString: -X -Y
408 ! Careful here: sometimes works, sometimes does not depending on server!?:
409 *cUImageviewprops*omOrient*labelString: -X -Y

410 !!!!!
411 ! For some reason the class defaults are used, so be careful below!
412 !!!!!
413 CUlom2.om9.labelString: No
414 ! Careful here: sometimes works, sometimes does not depending on server!?:
415 *cUImageviewprops*omUseAspect*labelString: No

416 *cUImageviewprops*lbSatColor.labelString: Saturated pixel color:
417 *cUImageviewprops*lbSatValue.labelString: Saturated pixel cutoff:
418 *cUImageviewprops*lbBoxSize.labelString: Integration box size:
419 *cUImageviewprops*lbReserve.labelString: Reserved:
420 *cUImageviewprops*lbRescale.labelString: Auto rescale:
421 *cUImageviewprops*lbAspect.labelString: Pixel aspect ratio:
422 *cUImageviewprops*lbUseAspect.labelString: Preserve aspect ratio:
423 *cUImageviewprops*lbOrient.labelString: Orientation:
424 *cUImageviewprops*lbBlackPix.labelString: Black pixel in image:
425 *cUImageviewprops*lbWhitePix.labelString: White pixel in image:
426 *cUImageviewprops*lbMinMaxO.labelString: ?
427 *cUImageviewprops*lbAvgSDO.labelString: ?
428 *cUImageviewprops*lbMinMax.labelString: Min, Max:
429 *cUImageviewprops*lbAvgSD.labelString: Average, sd:
430 *cUImageviewprops*lbImgSize.labelString: Image size:
431 *cUImageviewprops*lbBorder.labelString: Image border fraction:

432 *cURefInviewprops*dialogTitle: d*TREK - Edit reflection view properties
433 *cURefInviewprops*lbRotRange.labelString: Rot range:
434 *cURefInviewprops*lbRestrictRange.labelString: Restrict range:
435 *cURefInviewprops*lbPredObs.labelString: Pred / Obsv:
436 *cURefInviewprops*lbVisible.labelString: Visible:
437 *cURefInviewprops*lbColor.labelString: Color:
438 *cURefInviewprops*lbSymbol.labelString: Symbol:
439 *cURefInviewprops*pushButton55.labelString: Yes
440 *cURefInviewprops*pushButton56.labelString: No
441 *cURefInviewprops*pushButton54.labelString: Predicted
442 *cURefInviewprops*pushButton57.labelString: Observed
443 *cURefInviewprops*pushButton60.labelString: Both
444 *cURefInviewprops*pushButton53.labelString: Yes
```

```
445 *cUIrefInviewprops*pushButton58.labelString: No
446 *cUIrefInviewprops*pushButton52.background: black
447 *cUIrefInviewprops*pushButton52.labelString: Black
448 *cUIrefInviewprops*pushButton59.background: white
449 *cUIrefInviewprops*pushButton59.labelString: White
450 *cUIrefInviewprops*pushButton64.background: red
451 *cUIrefInviewprops*pushButton64.labelString: Red
452 *cUIrefInviewprops*pushButton65.background: orange
453 *cUIrefInviewprops*pushButton65.labelString: Orange
454 *cUIrefInviewprops*pushButton66.background: yellow
455 *cUIrefInviewprops*pushButton66.labelString: Yellow
456 *cUIrefInviewprops*pushButton67.background: green
457 *cUIrefInviewprops*pushButton67.labelString: Green
458 *cUIrefInviewprops*pushButton68.background: blue
459 *cUIrefInviewprops*pushButton68.labelString: Blue
460 *cUIrefInviewprops*pushButton69.labelString: Cyan
461 *cUIrefInviewprops*pushButton70.labelString: Magenta
462 *cUIrefInviewprops*lbSize.labelString: Size:
463 *cUIrefInviewprops*pushButton21.labelString: + sign
464 *cUIrefInviewprops*pushButton22.labelString: x sign
465 *cUIrefInviewprops*pushButton50.labelString: o sign
466 *cUIrefInviewprops*pushButton51.labelString: [] box
467 *cUIrefInviewprops*lbRefInName.labelString: Reflection list:
468 !
469 !
470 *cUIPSfileSelect.dialogTitle: d*TREK - Select PostScript file name
471 *cUIPSfileSelect.selectionLabelString: PostScript file
472 *cUIPSfileSelect*lbComment.labelString: Comment
473 !
474 !
475 *cUIimagefileSelect*dialogTitle: Select image file and/or template name
476 *cUIimagefileSelect*selectionLabelString: Image file
477 *cUIimagefileSelect*okLabelString: OK
478 *cUIimagefileSelect*filterLabelString: Filter or Template
479 *cUIimagefileSelect*numlImages.sensitive: True
480 *cUIimagefileSelect*numlImages.scrollBar.minimum: 0
481 *cUIimagefileSelect*numlImages.scrollBar.maximum: 111
482 *cUIimagefileSelect*numlImages.textField.value: %.Of
483 *cUIimagefileSelect*numlImages.label.labelString: :::t"Num\"images"
484 *cUIimagefileSelect*seqIncr.sensitive: True
485 *cUIimagefileSelect*seqIncr.textField.value: %.Of
486 *cUIimagefileSelect*seqIncr.label.labelString: :::t"Seq\"incr\"
487 *cUIimagefileSelect*seqStart.sensitive: True
488 *cUIimagefileSelect*seqStart.textField.value: %.Of
489 *cUIimagefileSelect*seqStart.label.labelString: :::t"Seq\"start"

490 ! *cUIimagefileSelect*helpHTML: FileTemplate
491 *htmlcUIimagefileSelect: FileTemplate
```

In addition to the above resources, the following resources are also used. They are divided into a few broad categories: help configuration, html filenames, context-sensitive help lines, and error messages.

Help configuration

dtdisplay tries to have a fully user-configurable help facility. This is achieved by the following resources.

Resource	Default value	Notes
helpHTMLviewer	mosaic	Command to run the html viewer
helpDirectory	./	Directory for the html files
helpPrefix	"	Prefix to the html files
helpSuffix	.html	Suffix to the html files
helpDelimiter	" -- Help: "	Text placed between title and added help line
helpTmpDirectory	/tmp/	Temporary directory for mosaic control

The hyper-text markup language (html) viewer is a program external to dtdisplay such as NCSA mosaic. The default values used by the help facility can be changed by adding these resources to the application defaults file. dtdisplay runs the viewer in response to the user activating a help command. The actual html files are specified by an abbreviation with dtdisplay. The full html filenames are constructed from concatenating the translations of the helpDirectory, helpPrefix, the abbreviation and the helpSuffix together. Furthermore, the abbreviation can be changed through the use of a resource of the same name with the string html prepended. For example, if the abbreviation is FileTemplate (hard coded in the source code), then the help facility checks if the resource htmlFileTemplate is defined and if so, translates it to get the file basename. The name of the html file would be ./FileTemplate.html. If helpPrefix was changed to dtdisplay/ and the helpDirectory was changed to /usr/local/html then the file would be /usr/local/html/dtdisplay/FileTemplate.html. The installer of dtdisplay is responsible for putting the html files in a public location and changing the resource file to reflect this. dtdisplay defines the following resources for each of the **Help** buttons in the program:

```
htmlFileTemplate
htmlEditRefInView
htmlDloadRefIn
htmlViewZoom
htmlEditImageView
htmlFileOpen
htmlHelpHTML
htmlFileImageTemplate
htmlFileNewOverlay
htmlFileNewTile
```

The context-sensitive help line displayed in the window title may be overridden with the helpLine resource for push button, cascade button, toggle button, option menu, and text field widgets. The helpLine resource specifies a text string that will be placed in the main window title whenever a button is armed, menu is

mapped or (in Motif 1.2) a text field gains keyboard focus. There are default help lines in the dtdisplay program, but you can override them with a line such as:

```
*dtdisplay*pbFileTemplate.helpLine:    Pop-up the image template dialog
```

Error messages can also be overridden with through resources. The following are error message resources:

```
errorFileNewOverlay
```

Two colors are specified through resources. The first is the color of the text used for the pixel values when an image area is zoomed very large. The second is for the color of the plotted reflection list (until the Edit refln props dialog is implemented). The default values are shown next.

```
pixelValueColor:    blue  
reflnPlotColor:    red
```

Not every color is supported for the above resources. Currently, the allowed values are red, orange, yellow, green, blue, magenta, cyan, black and white.

5.2 Make

Source files for dtdisplay reside in two directories `.../DTREK/src/DTDISPLAY` and `.../DTREK/src/DTTREK`. Use the makefiles specific for your platform. The makefiles are called `Makefile.platform`, where *platform* describes succinctly the operating system (e.g. `sgi4`, or `hp`). Copy the one for your system to `Makefile` in each directory named below. Make the object library `libdtrek.a` from source files found in the `DTTREK` directory:

```
cd .../DTREK/src/DTTREK  
gmake lib
```

then change directory to `.../DTREK/src/DTDISPLAY` and type `gmake`:

```
cd ../DTDISPLAY  
gmake
```

Appendix

This appendix contains definitions for **image** and **scan** as used by d*TREK and dtdisplay.

Image An **image** is the output of a position-sensitive detector. It has a header that describes the image and how it was made. dtdisplay reads image headers to get information about images, so that users do not need to describe the image to the program.

Rotation A **rotation** describes the goniometer movement while an image is exposed. It has the following properties:

Rotation axis

The 3D vectorial representation of the rotation axis in world coordinates when the crystal goniometer is at the datum position. The axis goes from the crystal position (0,0,0) in the axis direction such that a right-hand rotation is the positive direction. For example, if the omega axis points along X, then Rotation axis is (1, 0, 0). General rotation axes may be easily specified this way. In addition, the crystal goniometer datum position specifies a rotation angle of 0 around this axis.

Start angle The angle of the rotation axis at the start of exposure

End angle The angle of the rotation axis at the end of exposure

angle **Increment** The difference between the Start angle and End angle

Time The exposure time in seconds of the image

Oscillations The number of times the axis oscillated from the Start angle to the End angle back to the Start angle. A simple rotation will have Oscillations equal to 0.

Dark of the The number of dark images collected before the start of rotation and used for non-uniformity correction.

DarkUpdate The dark image update interval. After this number of images, a new dark image should be made.

DarkLimit The dark image change limit. If the dark current changes by more than this fraction, then a new dark image should be collected.

DCoffset The number of DC offset images collected before the start of the rotation and used for non-uniformity correction.

DC Update The DC offset image update interval. After this number of images, a new DC offset image should be made. This should probably always be 0.

Scan A **scan** is a set of contiguous (in rotation angle) images with similar properties. The images are generated by a single detector so they have the same dimensions, exposure time, angular increment and non-uniformity of response. A scan has the same global properties as described for Rotation, but the Scan End Angle is the end angle of the last image in the scan. Other Scan properties which are not pertinent to dtdisplay are described in the dtfind documentation.