

**NAME**

xview – xview toolkit information

**SYNOPSIS**

There is no **xview** command per se, but this manual page will briefly describe XView features and functions.

**AVAILABILITY**

XView is available with the OpenWindows distribution.

**DESCRIPTION**

**XView** (X Window-System-based Visual/Integrated Environment for Workstations) is an Open Look user-interface toolkit which supports development of interactive, graphics-based applications running under the X Window System. For detailed information see the *XView Programming Manual* and the *XView Reference Manual*.

**USAGE****Compiling XView Programs**

XView programs are compiled with the following command line:

```
cc sourcefile.c -o outputfile -lxview -lolgx -lX11
```

**Generic XView Functions**

**xv\_init()** Establishes the connection to the server, initializes the Notifier and the Defaults/Resource-Manager database, loads the Server Resource-Manager database, reads any passed attributes, and installs a default X11 Errorhandler.

```
Xv_Server
xv_init(attrs)
    <attribute-value list> attrs;
```

**Note:** **attrs** is a NULL terminated attribute-value list.

**xv\_create()** Creates an object.

```
Xv_object
xv_create(owner, package, attrs)
    Xv_object owner;
    Xv_pkg package;
    <attribute-value list> attrs;
```

**xv\_destroy()** Destroys an object.

```
int
xv_destroy(object)
    Xv_opaque object;
```

**xv\_find()** Finds an object that meets certain criteria; or if the object doesn't exist, creates it (default behavior which can be defeated using **XV\_AUTO\_CREATE**, **FALSE**).

```
Xv_opaque
xv_find(owner, package, attrs)
    Xv_object owner;
    Xv_pkg package;
    <attribute-value list> attrs;
```

**xv\_get()** Gets the value of a single attribute.

```

Xv_opaque
xv_get(object, attrs)
  Xv_object object;
  <attribute-value list> attrs;

```

**xv\_set()** Sets the value of one or more attributes.

```

Xv_opaque
xv_set(object, attrs)
  Xv_object object;
  <attribute-value list> attrs;

```

### Internationalized Support

XView now has support for internationalization. This includes locale setting, localized text handling, and dynamic object layout. See the *XView Programming Manual* for details.

### Command Line Resource Arguments

XView-based applications display characteristics can be controlled by supplying command line arguments to the applications at start-up. The usage is as follows:

```
% program -argument1 value1 -argument2 value2...
```

In the tables below, **Argument(s)** shows the short argument followed by the long argument—either can be used. **Type** describes the type of value the arguments can receive. **Resource** describes the X resource name modified by the arguments. **Default** is the default value. **Description** describes what the arguments do. **Example** shows an example of a command using the argument.

**Argument(s):** -Wx, or -scale  
**Type:** string ("small", "medium", "large", "extra\_large")  
**Resource:** Window.Scale  
**Default:** medium  
**Description:** Sets the initial scale of the application (larger or smaller). **small** is 10 pixels, **medium** is 12 pixels, **large** is 14 pixels and **extra\_Large** is 19 pixels. The font.name resource will override the scale.  
**Example:** cmdtool -scale extra\_large

**Argument(s):** -Wt, -fn, or -font  
**Type:** string  
**Resource:** Font.Name  
**Default:** lucidasans-12  
**Description:** Sets the name of the font used for the application. Does not set the font for frame header and frame menu header. These are controlled by the window manager. To find out what fonts are available, use the **xlsfonts(1)** command. If the font you specify cannot be found, you will see an error message such as:

```

XView warning: Cannot load font 'galant-24' (Font package)
XView warning: Attempting to load font '-b&h-lucida-medium-r-normal-
sans-*-120-*-*-*-*-*' instead (Font package)

```

**Example:** cmdtool -fn fixed

**Argument(s):** -Ws, or -size  
**Type:** integer integer

**Resource:** Window.Width and Window.Height  
**Default:** depends on application  
**Description:** Sets the width and height of the application's base frame. The values are in pixels.  
**Example:** cmdtool -Ws 400 500

**Argument(s):** -Ww, or -width  
**Type:** int (number of columns)  
**Resource:** window.columns  
**Default:** None  
**Description:** Specifies the width, in columns, of the application.  
**Example:** cmdtool -width 40  
(starts a command tool 40 columns wide)

**Argument(s):** -Wh, or -height  
**Type:** int (number of columns)  
**Resource:** window.rows  
**Default:** None  
**Description:** Specifies the height, in rows, of the application.  
**Example:** cmdtool -height 40  
(starts a command tool 40 rows high)

**Argument(s):** -Wp, or -position  
**Type:** integer integer  
**Resource:** Window.X and Window.Y  
**Default:** depends on window manager  
**Description:** Sets the initial position of the application's base frame in pixels. The upper left corner of the screen is at position (0,0), with the x-axis increasing to the left, and the y-axis increasing downward. These values will also be generated by the "Save Workspace" option on the root menu into the **\$HOME/.openwin-init** file when using the Open Look Window Manager.  
**Example:** cmdtool -Wp 100 200

**Argument(s):** -WG, or -geometry  
**Type:** string of the format <width>x<height>{+-}<xoffset>{+-}<yoffset>  
**Resource:** Window.Geometry  
**Default:** depends on application and window manager  
**Description:** This sets both the size and the placement of the application's base frame. This option has priority over the **-size** and **-position** arguments. The size and placement parts of the value are optional. You can set just the size, just the position, or both. The size values are measured in pixels, and the position values use the same semantics as **-position**. However, if you use the '-' in front of an X value, it will be taken as relative to the right hand side of the screen, instead of the left. Likewise, if you use the '-' with the Y value, it will be taken relative to the bottom of the screen instead of the top.

**Examples:**

**cmdtool -geometry 500x600**  
(will make the base frame 500x600 pixels, with the position set by the window manager)

**cmdtool -WG +10+20**  
(will make the base frame of default size with the left hand side of the frame 10 pixels from the left hand side of the screen, and the top of the frame 20 pixels from the top of the screen)

**cmdtool -WG -10+20**  
(will make the base frame of default size with the right hand side of the frame 10 pixels from the right hand side of the screen, and the top of the frame 20 pixels from the top of the screen)

**cmdtool -geometry 400x300-0-0**

(will make the base frame 400x300 pixels with the right hand side of the frame flush against the right hand side of the screen, and the bottom of the frame flush with the bottom of the screen)

- Argument(s):** -WP, -icon\_position  
**Type:** integer integer  
**Resource:** Icon.X Icon.Y  
**Default:** depends on window manager  
**Description:** Sets the position of the application's icon in pixels. Uses the same semantics as **-position** for base frames.  
**Example:** cmdtool -WP 400 20
- Argument(s):** -Wl, -label, or -title  
**Type:** string  
**Resource:** Window.Header  
**Default:** N/A  
**Description:** Sets a default label for the base frame's header. However, the application can overwrite this setting and display its own header.  
**Example:** cmdtool -Wl "Header Text"
- Argument(s):** -Wi, and +Wi  
**Type:** boolean  
**Resource:** Window.Iconic  
**Default:** +Wi  
**Description:** Controls how an application will come up, open or closed (iconified).  
**Examples:** cmdtool +Wi (will make the cmdtool come up open)  
 cmdtool -Wi (will make the cmdtool come up closed)
- Argument(s):** -depth  
**Type:** integer  
**Resource:** Window.Depth  
**Default:** Depth of server's default visual  
**Description:** Specifies the depth of base frame. If this depth is not supported by the server, the default depth will be used instead. If this is specified in conjunction with -visual, then the exact visual will be used.  
**Example:** cmdtool -depth 4
- Argument(s):** -visual  
**Type:** string (one of the values: StaticGray, GrayScale, StaticColor, PseudoColor, TrueColor, or DirectColor).  
**Resource:** Window.Visual  
**Default:** Server's default visual  
**Description:** Specifies the visual class of the base frame. If this visual class is not supported by the server, the default visual class will be used instead. If this is specified in conjunction with -depth, then the exact visual will be used.  
**Example:** cmdtool -visual StaticGray
- Argument(s):** -Wf, or -foreground\_color  
**Type:** integer integer integer  
**Resource:** Window.Color.Foreground  
**Default:** 0 0 0  
**Description:** See Description in -Wb below.

**Argument(s):** -Wb, or -background  
**Type:** integer integer integer  
**Resource:** Window.Color.Background  
**Default:** 255 255 255  
**Description:** These options allow the user to specify the foreground color (e.g., the color of the text in a textsw), or the background color (e.g., the color that the text is painted on) of an application. The three values should be integers between 0 and 255. They specify the amount of red, green and blue that is in the color. See **-fg** and **-bg** below for information on similar functions.  
**Example:** cmdtool -Wf 0 0 255 -Wb 100 100 100  
(would come up with a blue foreground, with a gray background)

**Argument(s):** -fg, or -foreground  
**Type:** string (color name, or hexadecimal color specification)  
**Resource:** Window.Color.Foreground  
**Default:** black  
**Description:** See Description in -bg below.

**Argument(s):** -bg, or -background  
**Type:** string (color name, or hexadecimal color specification)  
**Resource:** Window.Color.Background  
**Default:** white  
**Description:** These options are similar to the -Wf and -Wb options, except that they take a color argument in the form of a predefined color name (lavender, grey, goldenrod, etc.) from **\$OPENWINHOME/lib/rbg.txt**, or a hexadecimal representation. The hexadecimal representation is of the form pound sign (#) followed by the hexadecimal representation of the red, green and blue aspects of the color.

**Examples:** cmdtool -fg blue -bg gray  
(comes up with a blue foreground, with a gray background)  
cmdtool -fg #d800ff -bg white  
(comes up with a purple foreground, with a white background)

**Argument(s):** -WI, or -icon\_image  
**Type:** string  
**Resource:** Icon.Pixmap  
**Default:** depends on application  
**Description:** Sets the default filename for the icon's image. However, the application can overwrite this setting and display its own icon image. The file must be in XView icon format. The program **iconedit(1)** will allow one to create an image in the icon format. Several icons have been provided in the directory **\$OPENWINHOME/include/images**. By convention, icon format files end with the suffix **.icon**.

**Example:** cmdtool -WI /usr/include/images/stop.icon

**Argument(s):** -WL, or -icon\_label  
**Type:** string  
**Resource:** Icon.Footer  
**Default:** depends on application  
**Description:** Sets a default label for the base frame's icon. However, the application can overwrite this setting and display its own icon label.

**Example:** cmdtool -WL "Icon Label"

**Argument(s):** -WT, or -icon\_font  
**Type:** string

**Resource:** Icon.Font.Name  
**Default:** depends  
**Description:** Sets the name of the font used for the application's icon.  
**Example:** cmdtool -WT '\*century schoolbook\*'

**Argument(s):** -Wd, or -default  
**Type:** string string  
**Resource:** given by the first string  
**Default:** none  
**Description:** This option allows the user to set resources that don't have command line equivalents. The format is **-default resource-name value** . The XView resources without specific command line arguments are discussed in the following section.  
**Example:** cmdtool -default OpenWindows.ScrollbarPlacement left

**Argument(s):** -xrm  
**Type:** string  
**Resource:** given in the string  
**Default:** none  
**Description:** This option allows the user to set resources that don't have command line equivalents. This is similar to the -default option, but it takes only one argument, a string in the form of resource-name:value.  
**Example:** cmdtool -xrm OpenWindows.ScrollbarPlacement:right

**Argument(s):** -WH, or -help  
**Type:** none  
**Resource:** none  
**Default:** none  
**Description:** Prints a description of the valid xview command line arguments for the application.

**Argument(s):** -sync or -synchronous, and +sync or +synchronous  
**Type:** boolean  
**Resource:** Window.Synchronous  
**Default:** +synchronous  
**Description:** These options allow you to make the connection that the application has with the X11 server either synchronous (-sync) or asynchronous (+sync).

**Argument(s):** -Wr, or -display  
**Type:** string (host:display{.screen})  
**Resource:** Server.Name  
**Default:** taken from the **DISPLAY** environment variable  
**Description:** Sets the name of the X11 server on which to connect. **host** is the name or address of the machine on whose server you have permission to display. **display** is a number corresponding to the server on which to display for that machine, and **screen** corresponds to which screen for the server. See reference manual page on **xhost(1)** for more details on adding to permissions list.  
**Examples:** cmdtool -display foobar:0  
 (will bring up a cmdtool on the default screen of the display #0 of host foobar)  
 cmdtool -display foobar:0.1  
 (will bring up a cmdtool on screen #1 of display #0 of host foobar)

**Argument(s):** -Wdxio, or -disable\_xio\_error\_handler  
**Type:** boolean  
**Resource:** none

<b>Default:</b>	enable xio handler—this option disables it
<b>Description:</b>	This option is useful for debugging an application. Whenever there is a fatal XIO error, the server will print an error message before exiting. XView installs a error handler to keep those messages from appearing. If you would like to see these messages, use this option.
<b>Argument(s):</b>	-rv (or -reverse), and +rv (or +reverse)
<b>Type:</b>	boolean
<b>Resource:</b>	Window.ReverseVideo
<b>Default:</b>	False
<b>Description:</b>	These options control whether the foreground and background colors of the application will be reversed. If True, the foreground and background colors will be swapped. The -rv flag will set this to True, while the +rv will set it to False. This is really only useful on monochrome displays.
<b>Argument(s):</b>	-name
<b>Type:</b>	string
<b>Resource:</b>	None
<b>Default:</b>	argv[0]
<b>Description:</b>	Specifies the instance name of the application. This name is used to construct the resource name used to perform lookups in the X11 Resource Manager to look for the values of customizable attributes.

### Internationalized Command Line Resource Arguments

The following command line arguments are relevant to internationalization. Locale refers to the language and cultural conventions used in a program. Locale setting is the method by which the language and cultural environment of a system is set. Locale setting affects the display and manipulation of language-dependent features.

The internationalization features that XView now supports include locale setting. One of the ways locale can be set is with command line options. See the *XView Programming Manual* for details on other methods.

<b>Argument(s):</b>	-lc_basicalocale
<b>Type:</b>	string
<b>Resource:</b>	basicLocale
<b>Default:</b>	"C"
<b>Description:</b>	Specifies the basic locale category, which sets the country of the user interface.
<b>Argument(s):</b>	-lc_displaylang
<b>Type:</b>	string
<b>Resource:</b>	displayLang
<b>Default:</b>	"C"
<b>Description:</b>	Specifies the display language locale category, sets the language in which labels, messages, menu items, and help text are displayed.
<b>Argument(s):</b>	-lc_inputlang
<b>Type:</b>	string
<b>Resource:</b>	inputLang
<b>Default:</b>	"C"
<b>Description:</b>	Specifies the input language locale category, sets the language used for keyboard input.
<b>Argument(s):</b>	-lc_numeric
<b>Type:</b>	string

**Resource:** numeric  
**Default:** "C"  
**Description:** Specifies the numeric locale category, which defines the language used to format numeric quantities.

**Argument(s):** -lc\_timeformat  
**Type:** string  
**Resource:** timeFormat  
**Default:** "C"  
**Description:** Specifies the time format locale category, which defines the language used to format time and date.

### Command Line Options/X Resources for Debugging

The following switches/resources can be used during development to avoid the locking up of screens or other effects of X grabs that are done by XView.

It should be noted that these options/resources should only be used by developers and are not for normal usage. The X grabs are done for a specific reason, and are not meant to be customizable. Without the X grabs, certain features in XView (those that depend on X grabs) might not function properly.

**Argument(s):** -Wfsdb, or -fullscreendebg  
**Type:** boolean  
**Resource:** Fullscreen.Debug  
**Default:** FALSE  
**Description:** Enables/disables fullscreen debugging mode during which X grabs (XGrabServer(), XGrabKeyboard(), XGrabPointer()) are not done. When using the fullscreen pkg, the X11 server will be grabbed which prevents other windows on the server from responding until the grab has been released by the one window which initiated the grab. Refer to Appendix F in the *XView Manual: Converting SunView Applications* for further details.

**Argument(s):** -Wfsdb, or -fullscreendebgserver  
**Type:** boolean  
**Resource:** Fullscreen.Debugserver  
**Default:** FALSE  
**Description:** Enables/disables server grabbing (XGrabServer()) that is done via the fullscreen pkg. Refer to the Appendix F in the *XView Manual: Converting SunView Applications* for further details.

**Argument(s):** -Wfsdbk, or -fullscreendebgkbd  
**Type:** boolean  
**Resource:** Fullscreen.Debugkbd  
**Default:** FALSE  
**Description:** Enables/disables keyboard grabbing (XGrabKeyboard()) that is done via the fullscreen pkg. Refer to the Appendix F in the *XView Manual: Converting SunView Applications* for further details.

**Argument(s):** -Wfsdbp, or -fullscreendebgptr  
**Type:** boolean  
**Resource:** Fullscreen.Debugptr  
**Default:** FALSE  
**Description:** Enables/disables pointer grabbing (XGrabPointer()) that is done via the fullscreen pkg. Refer to the Appendix F in the *XView Manual: Converting SunView Applications* for further details.

**Argument(s):** -Wdpgs, or -disable\_pass\_grab\_select  
**Type:** boolean  
**Resource:** Window.PassiveGrab.Select  
**Default:** TRUE  
**Description:** Disables the passive grab that is done on the SELECT button. XView does a passive grab on the SELECT button to avoid input focus race conditions. When this passive grab is disabled, input focus race conditions may be seen.  
**Example:** % cmdtool -disable\_pass\_grab\_select  
 This executes a cmdtool that does not perform any passive grabs on the SELECT button. To do the same thing using X resources, add the following entry to the X resource database:  
 Window.PassiveGrab.Select:False

### .Xdefaults File

The **.Xdefaults** file is used to store and retrieve resource settings. We recommend, however, that you use the command line arguments described above in order to change display characteristics. Changing the resources in the **.Xdefaults** file will modify the behaviour of the user's session. Novice users should not casually hand modify these settings. Before attempting edits to this file please read the appropriate sections of the *Xlib Programming Manual* on the file format and the specific properties you intend to change.

Note that resources documented below do not have command line arguments. It is still possible, however, to change them without altering the **.Xdefaults** file. Refer to the command line arguments **-xrm** and **-defaults** for instructions on how to do this. Additional resources that have command line arguments are documented in the previous section. For mouseless resources refer to the *XView Programming Manual*.

The resources are documented in the following format:

**Resource:** *Resource Name ( If the resource can be modified by the OpenWindows Property Sheet, the word **Props** will be present.)*  
**Values:** *Possible Values, and/or Format of Values to be Assigned to Resource (Default Value)*  
**Description** *Description of Resource.*

**Resource:** window.synchronous, +sync -sync  
**Values:** True, False (False)  
**Description** Useful when debugging or tracking down a problem since the error codes emitted from Xlib will correspond to the immediate request made. Running in a synchronous mode will cause the application to run significantly slower.

**Resource:** mouse.modifier.button2  
**Values:** Shift, Ctrl, any valid modifier keysym (Shift)  
**Description** When using a mouse with less than three buttons, this resource gets an equivalent mapping for the second button which is the ADJUST button on a three button mouse. For more information on keysyms, see the **xmodmap(1)** reference manual page, Xlib documentation, and the include file **\$OPENWINHOME/include/X11/Xkeymap.h**.

**Resource:** mouse.modifier.button3  
**Values:** Shift, Ctrl, any valid modifier keysym (Ctrl)  
**Description** When using a mouse with less than three buttons, this resource gets an equivalent mapping for the third button which is the MENU button on a three button mouse. For more information on keysyms, see the **xmodmap** reference manual page, Xlib documentation, and the include file **\$OPENWINHOME/include/X11/Xkeymap.h**.

**Resource:** OpenWindows.beep (Props)

<b>Values:</b>	never, notices, always (always)
<b>Description:</b>	When the value is <b>notices</b> , the audible bell will ring only when a notice pops up. When the value is <b>never</b> , the audible bell will never ring. When the value is <b>always</b> , the audible bell will always ring when the bell function is called by a program.
<b>Resource:</b>	alarm.visible
<b>Values:</b>	True, False (True)
<b>Description:</b>	When ringing the bell in an XView program, flash the window as well to alert the user.
<b>OpenWindows.windowColor (Props)</b>	
<b>Values:</b>	any valid X11 color specification (#cccccc—80% grey)
<b>Description:</b>	Specify the base color for control areas for 3-D look. Takes hexadecimal representation. Three other colors used for shading and highlighting are calculated based upon the value of the specified control color. The actual calculated values are done by the OLGX library to provide a consistent color calculation between XView and OLWM. The desktop properties program allows a full range of customization and previews what the chosen 3-D look will look like. Does not apply to monochrome displays.
<b>Resource:</b>	OpenWindows.workspaceColor (Props)
<b>Values:</b>	any valid X11 color specification (#cccccc—80% grey)
<b>Description:</b>	Specifies the color for the root window and the background color for icons that blend into the desktop.
<b>Resource:</b>	xview.iccmmcompliant
<b>Values:</b>	True, False (True)
<b>Description:</b>	When False, tells XView to set window manager hints in a way that was used before the ICCCM was adopted. Useful for window managers that were released before X11R4. Not needed with the Open Look Window Manager provided with Open Windows.
<b>Resource:</b>	OpenWindows.3DLook.Color
<b>Values:</b>	True, False (True on all but monochrome screens)
<b>Description:</b>	When False, do not use the 3-D look on a color or greyscale screen.
<b>Resource:</b>	OpenWindows.dragRightDistance (Props)
<b>Values:</b>	N (100)
<b>Description:</b>	Used by menus to determine when a pullright submenu would display when dragging over the menu item near a submenu. N is an integer greater than 0. A reasonable value might start at 20 and go to 200 or so. May need to try different values to see what feels best to each person.
<b>Resource:</b>	Selection.Timeout
<b>Values:</b>	N (3)
<b>Description:</b>	Selection timeout value. N indicates the number of seconds that a requestor or a selection owner waits for a response.
<b>Resource:</b>	OpenWindows.GotoMenu.RecentCount
<b>Values:</b>	integer between 0 and 15 (8)
<b>Description:</b>	Specifies the number of recently visited directories shown in the Go To Menu of a File Chooser.
<b>Resource:</b>	OpenWindows.GotoMenu.UserDirs
<b>Values:</b>	string-list (NULL)
<b>Description:</b>	new-line (0 separated list of full-path names to directories that is added to the top of the Go To Menu of a File Chooser.

- Resource:** OpenWindows.KeyboardCommand.\*  
**Description:** These resources determine mouseless semantic action and its corresponding key binding. Refer to the *XView Reference Manual* for a complete listing and explanation of the OpenWindows.KeyboardCommand.\* resources. Refer to the *XView Programming Manual* for information on the mouseless model.
- Resource:** OpenWindows.KeyboardCommands  
**Values:** SunView1, Basic, or Full  
**Description:** Controls the level of mouseless operation. All of the OpenWindows.KeyboardCommand resource mappings may be modified by users, or by specifying one the the three values for OpenWindows.KeyboardCommands. For detailed information see the *XView Programming Manual*
- Resource:** OpenWindows.MenuAccelerators  
**Values:** True or False (True)  
**Description:** Specifies whether or not to activate all keyboard menu acceleration defined by applications. Menu accelerators are keystrokes that can be used to invoke menu commands directly. They can be seen on the right side of frequently used menu items as a set of keyboard qualifiers (with a graphical diamond mark representing the meta key) and an accelerator key.
- Resource:** OpenWindows.MouseChordMenu  
**Values:** True, False (False)  
**Description:** Turns on the mouse chording mechanism. Mouse chording was implemented to allow XView to work with two-button mice. Pressing the SELECT and the ADJUST buttons at the same time will act as MENU button.
- Resource:** OpenWindows.MouseChordTimeout  
**Values:** N (100)  
**Description:** Mouse chording time-out value. N is in micro-seconds.
- Resource:** OpenWindows.SelectDisplaysMenu (Props)  
**Values:** True, False (False)  
**Description:** When True, the SELECT button (usually left mouse) will display the menu as well as the MENU button (usually right mouse).
- Resource:** OpenWindows.popupJumpCursor (Props)  
**Values:** True, False (False)  
**Description:** When False, do not warp the mouse to the notice when it appears.
- Resource:** notice.beepCount  
**Values:** N (1)  
**Description:** Where N is an integer to specify how many times to ring the bell when a notice appears.
- Resource:** OpenWindows.scrollbarPlacement (Props)  
**Values:** Left, Right (Right)  
**Description:** When set to **Left**, put all scrollbars on the lefthand side of the window or object.
- Resource:** OpenWindows.multiClickTimeout (Props)  
**Values:** N (4)  
**Description:** Where N is an integer greater than 2. Set the number of tenths of a second between clicks for a multi-click. A click is button-down, button-up pair.

- Resource:** text.delimiterChars  
**Values:** string (' \011!\"#\$%&\'()\*+,-./:;<=>?@[\\]^\_`{|}~')  
**Description:** This resource allows the user to select the delimiter characters that are used when doing word level selections in the XView package. It was added because of the needs of the international marketplace, and it allows the user to define the local delimiters for the character set that is being used with the current keyboard and Sun workstation.
- Note that the octal characters can be scrambled by **Xrm** during a rewrite of the value of text.delimiterChars. **Xrm** interprets the text.delimiterChar string when it is loaded. Specifically it will decode the backslashed portions of the string and convert them to octal representations. When this is passed to the client application, the logic will function correctly. However, this misbehavior of **Xrm** causes the string to be stored incorrectly if the user saves the **.Xdefaults** file using the **Xrm** content of the string. The specific problem(s) that occur are the stripping of the backslash characters and the expansion of the tab character (\011).
- To correct this problem, one can put the text.delimiterChar entry into an **.Xdefaults** file that will not be overwritten when saving the workspace properties (for example, a system wide defaults file). Or a copy of the text.delimiterChar entry can be inserted after **.Xdefaults** file saves.
- Resource:** scrollbar.jumpCursor (Props)  
**Values:** True, False (True)  
**Description:** When False, the scrollbar will not move the mouse pointer when scrolling.
- Resource:** scrollbar.repeatDelay  
**Values:** N (100)  
**Description:** Where N is some integer greater than 2. Specifies the time in milliseconds when a click becomes a repeated action.
- Resource:** scrollbar.pageInterval  
**Values:** N (100)  
**Description:** Where N is some integer greater than 2. Specifies the time in milliseconds between repeats of a single page scroll.
- Resource:** scrollbar.lineInterval  
**Values:** N (1)  
**Description:** Where N is some integer greater than 0. Specifies the time in milliseconds between repeats of a single line scroll. How long to pause scrolling when holding down the SELECT button on the scrollbar elevator. Scrollbar sets up a timer routine for repeats.
- Resource:** text.maxDocumentSize  
**Values:** N (2000)  
**Description:** Where N specifies the bytes used in memory before a text file is saved to a file on disk. Once this limit is exceeded, the text package will send a notice to the user to tell them that no more insertions are possible. If the file being edited is saved to a file, or it is a disk file being edited, then the limit does not apply.
- Resource:** text.retained  
**Values:** True, False (False)  
**Description:** If True, retain text windows with server backing store.
- Resource:** text.extrasMenuFilename

<b>Values:</b>	filename ( <code>/usr/lib/text_extras_menu</code> )
<b>Description</b>	Where filename is an absolute location to a file. Can also be set via environment variable EXTRASMENU. This file is used for the text package's Extras menu. The commands specified in the extras menu are applied to the contents of the current selection in the textsw window and then it inserts the results at the current insertion point.
<b>Resource:</b>	text.enableScrollbar
<b>Values:</b>	True, False (True)
<b>Description</b>	When False, do not put a scrollbar on textsw objects.
<b>Resource:</b>	text.againLimit
<b>Values:</b>	N (1)
<b>Description</b>	Where N is an integer between 0 and 500. Number of operations the "again history" remembers for a textsw.
<b>Resource:</b>	text.autoIndent
<b>Values:</b>	True, False (False)
<b>Description</b>	When True, begin the next line at the same indentation as the previous line as typing in text.
<b>Resource:</b>	text.autoScrollBy
<b>Values:</b>	N (1)
<b>Description</b>	Where N is an integer between 0 and 100. Specifies the number of lines to scroll when type-in moves insertion point below the view.
<b>Resource:</b>	text.confirmOverwrite
<b>Values:</b>	True, False (True)
<b>Description</b>	When False, do not give user confirmation if a save will overwrite an existing file.
<b>Resource:</b>	text.displayControlChars
<b>Values:</b>	True, False (True)
<b>Description</b>	When False, use an up arrow plus a letter to display the control character instead of the character that is available for the current font.
<b>Resource:</b>	Text.DeleteReplacesClipboard
<b>Values:</b>	True, False (False)
<b>Description</b>	This resource controls whether text that has been selected and then deleted by the delete key or replaced by any other keystroke will be copied to the clipboard. If the value is True, then the selected text will be copied to the clipboard. If the value is False, then the text selected does not replace the clipboard.  This resource also applies to the text selected for the filter function. If the resource is True, then the text selected for a filter function will replace the clipboard when the filter successfully finishes. If the resource is False, then the text selected does not replace the clipboard.
<b>Resource:</b>	text.undoLimit
<b>Values:</b>	N (50 maximum of 500)
<b>Description</b>	Where N is an integer between 0 and 500. How many operations to save in the undo history log. These operations will be undone when you press the "Undo" key in the text window.
<b>Resource:</b>	text.insertMakesCaretVisible

<b>Values:</b>	If_auto_scroll (Always)
<b>Description</b>	Controls whether insertion causes repositioning to make inserted text visible.
<b>Resource:</b>	text.lineBreak
<b>Values:</b>	Clip, Wrap_char, Wrap_word (Wrap_word)
<b>Description</b>	Determines how the textsw treats file lines when they are too big to fit on one display line.
<b>Resource:</b>	text.margin.bottom
<b>Values:</b>	N (0)
<b>Description</b>	Where N is an integer between -1 and 50. Specifies the minimum number of lines to maintain between insertion point and bottom of view. A value of -1 turns auto scrolling off.
<b>Resource:</b>	mouse.multiclick.space
<b>Values:</b>	N (4)
<b>Description</b>	Where N is an integer between 2 and 500. Specifies the maximum number of pixels between successive mouse clicks to still have the clicks considered as a multi-click event.
<b>Resource:</b>	text.storeChangesFile
<b>Values:</b>	True, False (True)
<b>Description</b>	When False, do not change the name of the current file being edited to the name of the file that is stored. The name of the current file is reflected in the titlebar of the textedit frame.
<b>Resource:</b>	text.margin.top
<b>Values:</b>	N (2)
<b>Description</b>	Where N is an integer between -1 and 50. Specifies the minimum number of lines to maintain between the start of the selection and the top of the view. A value of -1 means defeat normal actions.
<b>Resource:</b>	text.margin.left
<b>Values:</b>	N (8)
<b>Description</b>	Where N is an integer between 0 and 2000. Specifies the margin in pixels that the text should maintain between the left hand border of the window and the first character on each line.
<b>Resource:</b>	text.margin.right
<b>Values:</b>	N (0)
<b>Description</b>	Where N is an integer between 0 and 2000. Specifies the margin in pixels that the text should maintain between the right hand border of the window and the last character on each line.
<b>Resource:</b>	text.tabWidth
<b>Values:</b>	N (8)
<b>Description</b>	Where N is an integer between 0 and 50. Specifies the width in characters of the tab character.
<b>Resource:</b>	Text.LineSpacing
<b>Values:</b>	N (0)
<b>Description</b>	Where N is an integer which is the percentage of the maximum height of a character in the Textsw window font to use as interline spacing. Setting Text.LineSpacing to a nonzero positive number will increase the size of a Textsw proportionally. xv_set() of

WIN\_ROWS will still yield the correct number of rows. However, the window will be taller as compared to a Textsw with Text.LineSpacing set to 0. This resource allows XView to conform to TUV requirements. To meet TUV requirements, set Text.LineSpacing to 15 or greater.

<b>Resource:</b>	term.boldStyle
<b>Values:</b>	None, Offset_X, Offset_Y, Offset_X_and_Y, Offset_XY, Offset_X_and_XY, Offset_Y_and_XY, Offset_X_and_Y_and_XY, Invert (Invert)
<b>Description</b>	Specify the text bolding style for a terminal based window.
<b>Resource:</b>	term.inverseStyle
<b>Values:</b>	Enable, Disable, Same_as_bold (Enable)
<b>Description</b>	Specify the text inverting style for a terminal based window.
<b>Resource:</b>	term.underlineStyle
<b>Values:</b>	Enable, Disable, Same_as_bold (Enable)
<b>Description</b>	Specify the text underlining style for a terminal based window.
<b>Resource:</b>	term.useAlternateTtyswrc
<b>Values:</b>	True, False (True)
<b>Description</b>	When True, and a <b>\$HOME/.ttyswrc</b> is not found, look for an alternate <b>ttyswrc</b> file. When False, do not look for an alternate file one is not found in the home directory, <b>\$HOME/.ttyswrc</b> .
<b>Resource:</b>	term.alternateTtyswrc
<b>Values:</b>	filename ( <b>\$XVIEWHOME/lib/.ttyswrc</b> )
<b>Description</b>	Where filename specifies a complete filename and absolute path of an alternate <b>ttyswrc</b> file. This is only used if a <b>.ttyswrc</b> file is not found in <b>\$HOME/.ttyswrc</b> and term.useAlternateTtyswrc is True.
<b>Resource:</b>	term.enableEdit
<b>Values:</b>	True, False (True)
<b>Description</b>	When False, do not keep an editlog of what has been typed into the term window. This is set to false automatically when switching from a scrollable term to one that is not scrollable.
<b>Resource:</b>	ttysw.eightBitOutput
<b>Values:</b>	True, False (True)
<b>Description</b>	This resource controls whether characters modified by the meta modifier are encoded as eight-bit characters when passed to the ttysw's pty or are delivered as seven-bit characters.
<b>Resource:</b>	ttysw.yieldModifiers
<b>Values:</b>	Meta, Alt (The default is to not remove any semantic meaning from any modifiers)
<b>Description</b>	This resource takes as a value a list of modifier keys. Any semantic meaning (mouseless command or keyboard accelerator) that would normally be associated with the listed modifiers when the keyboard focus is in a ttysw or termsw would be removed.

## ENVIRONMENT

**\$OPENWINHOME** is the directory in which the server's directory hierarchy is installed.

**\$DISPLAY** is the name of the server and screen to which applications should display.

**\$LD\_LIBRARY\_PATH** is the SunOS shared library search path.

**\$HELPPATH** is the path that applications will search for Open Look Help files.

**FILES**

**\$OPENWINHOME/include/images**  
XView images

**\$OPENWINHOME/lib**  
XView Libraries

**\$OPENWINHOME/include**  
Include files

**\$OPENWINHOME/bin**  
Binaries

**\$OPENWINHOME/share/src/xview/demos**  
XView demo programs

**\$OPENWINHOME/share/src/xview/examples**  
XView example programs

**SEE ALSO**

**openwin(1), xnews(1), xlsfonts(1), xmodmap(1), iconedit(1)**