6
Command-line Arguments and XView Resources

This section lists XView properties that can be set with command-line options or by specifying values for resources. Values for most properties can be overridden programatically using XView attributes. For example, the position and size of an application's base frame can be set using the command-line option `–geometry`, or by changing the value of the `Window.Geometry` resource. However, if the application sets its size and position by setting attributes with `xv_set()`, the values specified in the call to `xv_set()` take precedence.

This section is divided into two parts: 6.1, which lists the properties that have an explicit command-line option, and 6.2, which lists the properties that do not have an explicit command-line option.

XView properties may be set in a number of ways, as the following list indicates. The list shows precedence for setting property values, from highest to lowest.

1. Calling `xv_set()` for one or more attributes that change the value of a property.
2. Specifying a flag name plus its value, if any, on the command line.
3. Editing or adding values for resource defaults in a *~/.Xdefaults* file.
4. Calling `xv_create()` for an object's attributes.
5. Using the package default values for the property.

From this list, you can see that command-line options have precedence over values set during `xv_create()`, but that all values are overridden by an explicit call to `xv_set()`.

The command-line options `–rv` (reverse), `–scale`, `–font`, `–foreground_color`, `–background_color`, and `–icon_font` apply to all top-level frames in the application. The remaining options apply only to the first top-level frame created by the application. Child subframes inherit properties from their parent frames. Therefore, command-line options eventually propagate to subframes unless the options are overridden programatically.

6.1 Command-line Options with Resources

–background

This option takes a single argument that is in the form of a predefined color name (lavender, gray, goldenrod, etc.) from *$OPENWINHOME/lib/rbg.txt* or a hexidecimal representation. The hexidecimal representation is of the form pound sign (#) followed by the hexidecimal representation of the red, green, and blue aspects of the color. Also see `-Wb`.

Short:	`–bg`
Type:	string (color name, or hexidecimal color specification)
Resource:	`Window.Color.Background`
Default:	white
Examples:	

```
cmdtool -foreground blue -background gray
```

Provides a blue foreground, with a gray background.

```
cmdtool -foreground #d800ff -background white
```

Provides a purple foreground, with a white background.

–default

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.

Short:	`–Wd`
Type:	string string
Resource:	Given by the first string
Default:	None
Example:	

```
cmdtool -default
OpenWindows.ScrollbarPlacement left
```

–defeateventsecurity

Enables an XView application to receive all synthethic events (sent via `XSendEvent()`).

Short:	`–WS`
Type:	boolean
Resource:	none
Default:	FALSE
Example:	

```
cmdtool -WS
```

–depth

Specifies the window's depth.

Type:	int
Resource:	`Window.Depth`

–disable_retained

This option is useful for applications running on a monochrome display, where server memory is at a minimum. For performance reasons, monochrome windows are by default retained by the server. Using retained windows will use more memory in the X11 server; however, it also speeds up repainting when the window is covered and uncovered by other windows. When `-disable_retained` is set, monochrome windows are not retained, thus saving server memory.

Short:	`–Wdr`
Type:	Boolean
Resource:	`Window.Mono.DisableRetained`
Default:	Not retained on color systems. Retained on monochrome systems.

–disable_xio_error_handler

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 an error handler to keep those messages from appearing. If you would like to see these messages, use this option.

Short:	`–Wdxio`
Type:	Boolean
Resource:	None
Default:	Enable XIO handler. Setting this option disables the XIO handler.

–display

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 manual page on `xhost` for more details on adding to permissions list.

Short:	`–Wr`
Type:	String of the form: `host:display{.screen}`
Resource:	`Server.Name`
Default:	Taken from the `DISPLAY` environment variable.
Example:	

```
cmdtool -display foobar:0
```

Brings up a `cmdtool` on the default screen of the display #0 on the host foobar.

```
cmdtool -display foobar:0.1
```

Brings up a `cmdtool` on screen #1 of display #0 of host foobar.

Command-line Arguments and XView Resources

–font, **–fn**

Sets the name of the font used for the application (not control areas). To find out what fonts are available, use the *xlsfonts* command.

Short:	`–Wt`
Type:	string
Resource:	`Font.Name` or `OpenWindows.RegularFont`
Default:	lucidasans-12
Example:	

```
cmdtool -font fixed
```

If the font you specify cannot be found, an error message is shown. For example:

```
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)
```

–foreground

This option specifies the foreground color. For example, an application's text in its textsw would take on the foreground color. Also see the `-background` option for information on similar functions. This option takes a single argument that is in the form of a predefined color name (lavender, gray, goldenrod, etc.) from *$OPENWINHOME/lib/rbg.txt* or a hexidecimal representation. The hexidecimal representation is of the form pound sign (#) followed by the hexidecimal representation of the red, green, and blue aspects of the color.

Short:	`–fg`
Type:	string (color name, or hexidecimal color specification)
Resource:	`Window.Color.Foreground`
Default:	black
Example:	

```
cmdtool -fg blue
```

Comes up with a blue foreground.

–foreground_color

This option allows the user to specify the foreground color of an application. It takes three values that should be integers between 0 and 255. They specify the amount of red, green, and blue that is in the color.

Short:	`–Wf`
Type:	integer integer integer
Resource:	`Window.Color.Foreground`
Default:	0 0 0
Example:	

```
cmdtool -Wf 0 0 255
```

Comes up with a blue foreground.

–fullscreendebug

Enables or disables fullscreen debugging mode during which XGrabs (`XGrabServer()`, `XGrabKeyboard()`, `XGrabPointer()`) are not done. When using the `FULLSCREEN` package, the X11 server is normally grabbed. This prevents other windows on the server from responding until the grab has been released by the one window which initiated the grab. Refer to the manual *Converting SunView Applications* for further details.

Short:	`–Wfsdb`
Type:	Boolean
Resource:	`Fullscreen.Debug`
Default:	`FALSE`

–fullscreendebugkbd

Enables or disables keyboard grabbing using `XGrabKeyboard()` that is done via the `FULLSCREEN` package.

Short:	`–Wfsdbk`
Type:	Boolean
Resource:	`Fullscreen.Debugkbd`
Default:	`FALSE`

–fullscreendebugptr

Enables or disables pointer grabbing `XGrabPointer()` that is done via the `FULLSCREEN` package. Refer to Appendix F of the manual *Converting SunView Applications* for further details.

Short:	`–Wfsdbp`
Type:	Boolean
Resource:	`Fullscreen.Debugptr`
Default:	`FALSE`

–fullscreendebugserver

Enables or disables server grabbing using `XGrabServer()` that is done with the `FULLSCREEN` package. Refer to Appendix F in the manual *Converting SunView Applications* for further details.

Short:	`–Wfsdbs`
Type:	Boolean
Resource:	`Fullscreen.Debugserver`
Default:	`FALSE`

–geometry

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 – (minus) 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 – (minus) with the *y* value, it will be taken relative to the bottom of the screen instead of the top.

Short: –WG
Type: string of the format: {WxH}{[+|-]x[+|-]y}
Resource: `Window.Geometry`
Default: Depends on the application.
Examples:

```
cmdtool -geometry 500x600
```

Makes the base frame 500×600 pixels, with the position set by the window manager.

```
cmdtool -WG +10+20
```

Makes the base frame of the 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
```

Makes the base frame of the 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
```

Makes the base frame 400×300 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.

–help

Prints a description of the valid command-line arguments for the application.

Short: –WH
Type: None
Resource: None
Default: None

–icon_font

Sets the name of the font used for the application's icon. To find out what fonts are available, use *xlsfonts*.

Short: –WT
Type: string
Resource: `Icon.Font.Name`
Default: Depends on the application.
Example:

```
cmdtool -WT '*century schoolbook*'
```

–icon_image

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` allows you to create an image in the icon format. Several icons are available in the directory *$OPENWINHOME/include/images*. By convention, icon format files end with the suffix *.icon*.

Short:	`-WI`
Type:	string
Resource:	`Icon.Pixmap`
Default:	Depends on the application.
Example:	

```
cmdtool -WI /usr/include/images/stop.icon
```


–icon_label

Sets a default label for the base frame's icon. However, the application can overwrite this setting and display its own icon label.

Short:	`-WL`
Type:	string
Resource:	`Icon.Footer`
Default:	Depends on the application.
Example:	

```
cmdtool -WL "Icon Label"
```

–icon_position

Sets the position of the application's icon in pixels. Uses the same semantics as `-position` for base frames.

Short:	`-WP`
Type:	integer integer
Resource:	`Icon.X Icon.Y`
Default:	Depends on the window manager.
Example:	

```
cmdtool -WP 400 20
```

–label

Sets a default label for the base frame's header. However, the application can overwrite this setting and display its own header.

Short:	`-Wl`
Type:	string
Resource:	`Window.Header`
Default:	Depends on the application.
Example:	

```
cmdtool -Wl "Header Text"
```

–lc_basiclocale

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 Version 3 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.

The `-lc_basiclocale` option specifies the basic locale category, which sets the country of the user interface.

Type:	string
Resource:	`basicLocale`
Default:	"C"

–lc_displaylang

Specifies the display language locale category, sets the language in which labels, messages, menu items, and help text are displayed.

Type:	string
Resource:	`displayLang`
Default:	"C"

–lc_inputlang

Specifies the input language locale category, sets the language used for keyboard input.

Type:	string
Resource:	`inputLang`
Default:	"C"

–lc_numeric

Specifies the numeric locale category, which defines the language used to format numeric quantities.

Type:	string
Resource:	`numeric`
Default:	"C"

–lc_timeformat

Specifies the time format locale category, which defines the language used to format time and date.

Type:	string
Resource:	`timeFormat`
Default:	"C"

–name

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 for customizable attributes.

Type:	string
Resource:	None
Default:	`argv[0]`

–position

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. To determine framebuffer size, one can use the `eeprom` command on the local machine. To determine screen size for a remote display, one can use Xlib functions (see the *Xlib Programming Manual*). 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.

Short:	`-Wp`
Resource:	`Window.X and Window.Y`
Type:	integer integer
Default:	Depends on the window manager.
Example:	

```
cmdtool -Wp 100 200
```

–scale

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. Any font resource overrides the scale unless the scale is specified on the command line.

Short:	`-Wx`
Type:	string
Valid Values:	{small, medium, large, extra_large}
Resource:	`Windows.Scale` or `OpenWindows.Scale`
Default:	"medium"
Example:	

```
cmdtool -scale extra_large
```

–size

Sets the width and height of the application's base frame. The values are in pixels.

Short:	`-Ws`
Type	integer integer
Resource:	`Window.Width and Window.Height`
Default:	Depends on the application.
Example:	

```
cmdtool -Ws 400 500
```

–synchronous, +synchronous

These options allow you to make the connection that the application has with the X11 server either synchronous (-sync) or asynchronous (+sync).

Short:	`-sync, +sync`
Type:	Boolean
Resource:	`Window.Synchronous`
Default:	`+synchronous`

–title Sets a default label for the base frame's header. However, the application can overwrite this setting and display its own header.

Short:	`-Wl`
Type:	string
Resource:	`Window.Header`
Default:	Depends on the application.
Example:	

```
cmdtool -Wl "Header Text"
```

–visual

Resource specifies the visual used for the window.

Type:	Visual
Resource:	`Window.visual`

–Wb This option allows the user to specify the background color (i.e., the color that text is painted on) for an application. The arguments are three values that should be integers between 0 and 255. They specify to the amount of red, green, and blue that is in the color.

Type:	integer integer integer
Resource:	`Window.Color.Background`
Default:	0 0 0

–Wd See the `–default` option.

–Wdr See the `–disable_retained` option.

–Wdxio

See the `–disable_xio_error_handler` option.

–Wf See the `–foreground_color` option.

–Wfsdb

See the `–fullscreendebug` option.

–Wfsdbs

See the `–fullscreendebugserver` option.

–WG See the `–geometry` option.

–WH See the `–help` option.

–Wi, and +Wi

These options control how an application will come up, open or closed (iconified).

Short:
Type: Boolean
Resource: `Window.Iconic`
Default: +Wi
Examples:

```
cmdtool +Wi
```

Makes `cmdtool` come up open.

```
cmdtool -Wi
```

Makes `cmdtool` come up closed.

–WI (uppercase i)
See the `–icon_image` option.

–Wl (lowercase L)
See the `–label` option.

–WL See the `–icon_label` option.

–Wp See the `–position` option.

–WP See the `–icon_position` option.

–Wr See the `–display` option.

–Ws See the `–size` option.

–Wt See the `–font` option.

–WT See the `–icon_font` option.

–Wx See the `–scale` option.

–xrm 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*.

Type:	String
Resource:	Given in the string
Default:	None
Example:	

```
cmdtool -xrm OpenWindows.ScrollbarPlacement:right
```

6.2 Additional Resources

The *.Xdefaults* file stores resource settings. We recommend that you use the command-line arguments in order to change display characteristics rather than changing the *.Xdefaults* file. Changing the resources in *.Xdefaults* modifies the behavior of the user's session. Novice users should not casually modify these settings.

Before attempting to edit the *.Xdefaults* 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 in this section do not have command-line arguments. It is possible to change these properties without altering the *.Xdefaults* file. The command-line arguments `-xrm` and `-defaults` provide instructions on how to specify values for any property. The resources that have command-line arguments are documented in the previous section.

Introduction.Resources (Props)

This is an example of the format for the resources described in this section. This field contains a brief description of the resource. If the resource can be modified by the OpenWindows Property Sheet, the resource name is followed by "(Props)".

Values: Val1, Val2 (Default). This field contains the possible values for the resource. If the resource may contain any value, the default(s) are provided. Defaults are shown enclosed in parentheses following the valid values.

alarm.visible

When ringing the bell in an XView program, flash the window as well to warn the user.

Values: True, False (True)

keyboard.cancel

Values: integer

keyboard.defaultAction

Values: integer

keyboard.inputFocusHelp

Values: integer

keyboard.nextElement

Values: integer

keyboard.deleteChar

Specifies the delete character. This resource applies to text windows only and not to panel text items. This would work in either `cmdtool` or `textedit` or the compose window of `mailtool`.

Values: C (177 = octal for Delete)
Where C is some character either typed into an editor or specified with an octal equivalent.

keyboard.deleteLine

Specifies the delete line character. This resource applies to text windows only and not to panel text items. This would work in either `cmdtool` or `textedit` or the compose window of `mailtool`.

Values: C
Where C is some character either typed into an editor or specified with an octal equivalent.

keyboard.deleteWord

Specifies the delete word character. This resource applies to text windows only and not to panel text items. This would work in either `cmdtool` or `textedit` or the compose window of `mailtool`.

Values: C (27 = octal for ˆW)
Where C is some character either typed into an editor or specified with an octal equivalent.

mouse.modifier.button2

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` reference manual page in the Xlib documentation, or the *include* file *$OPENWINHOME/include/X11/Xkeymap.h*.

Values: Shift, Ctrl, any valid modifier keysym (Shift)

mouse.modifier.button3

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*.

Values: Shift, Ctrl, any valid modifier keysym (Ctrl)

mouse.multiclick.space

Specifies the maximum number of pixels between successive mouse clicks to still have the clicks considered as a multi-click event.

Values: N (4)
Where N is an integer between 2 and 500.

notice.beepCount

Ringing the bell can consist of either an audible beep and/or a visual flash.

Values: N (1)
Where N is an integer to specify how many times to ring the bell when a notice appears.

notice.jumpCursor

Values: True, False

OpenWindows.3DLook.Monochrome

Values: True, False (True on all monochrome screens)

OpenWindows.3DLook.Color

When False, do not use the 3-D look on a color or grayscale screen.

Values: True, False (True on all but monochrome screens)

OpenWindows.beep (Props)

When the value is notices, the audible bell will ring only when a notice pops up. When the value is never, the audible bell will never ring. When the value is always, the audible bell will always ring when the bell function is called by a program.

Values: never, notices, always (always)

OpenWindows.BoldFont

Sets the font used by labels of a panel text item. If *-font* is specified on the command line, then it will override the font set by `OpenWindows.BoldFont`. To find out what fonts are available, use the *xlsfonts* command.

Values: font name

OpenWindows.dragRightDistance (Props)

Used by menus to determine when a pullright submenu would display when dragging over the menu item near a submenu.

Values: N (100)
N is an integer greater than 0. A reasonable value might start at 20 and go to 200.

OpenWindows.GotoMenu.UserDirs

User's directories in Goto Menu. <string-list> is a newline separated list of full-path directory names. Since the users home directory is always present it should not be in this list.

Values:	<string-list>
Default:	<null-string>
Example:	

```
OpenWindows.GotoMenu.UserDirs: /home/me/src0usr/openwin
```

OpenWindows.GotoMenu.RecentCount

Number of Recently Visited Directories Shown: where <count> is an non-negative integer which is the number of the recently visited directories shown in the Goto Menu. The default is 8, the maximum is 15.

Values:	<count>
Default:	<null-string>
Example:	

```
OpenWindows.GotoMenu.RecentCount:       10
```

Command-line Arguments and XView Resources

OpenWindows.KeyboardCommand.*

All of the `OpenWindows.KeyboardCommand` resource mappings may be modified by users, or by specifying one of three values for `OpenWindows.KeyboardCommands`: Sunview1 (which is the default), Basic, or Full. See the description for `OpenWindows.KeyboardCommands` and Chapter 6, *Handling Input*, in the *XView Programming Manual*, for more information.

OpenWindows.KeyboardCommand.Adjust

This mapping is loaded if `KeyboardCommands` is set to `Full`. This is a keyboard Core Functions resource.

Values: Insert+Alt

OpenWindows.KeyboardCommand.Again

This is a keyboard Core Functions resource.

Values: a+Meta, a+Ctrl+Meta, L2

OpenWindows.KeyboardCommand.Copy

This is keyboard Core Functions resource.

Values: u+Meta, L4

OpenWindows.KeyboardCommand.CopyThenPaste

This is keyboard Core Functions resource. This mapping is always loaded.

Values: p+Meta

OpenWindows.KeyboardCommand.Cut

This is keyboard Core Functions resource. This mapping is always loaded.

Values: x+Meta, L10

OpenWindows.KeyboardCommand.DataEnd

This mapping is always loaded. This is a keyboard Local Navigation command.

Default Values: End, R13, Return+Ctrl, End+Shift
Values (Basic or Full): End+Ctrl, R13+Ctrl

OpenWindows.KeyboardCommand.DataStart

Default Values: Home, R7, Return+Shift+Ctrl, Home+Shift
Values (Basic or Full): Home+Ctrl, R7+Ctrl

OpenWindows.KeyboardCommand.DefaultAction

This is Keyboard Core Functions resource. This mapping is always loaded.

Values: Return+Meta

OpenWindows.KeyboardCommand.Down

This mapping is always loaded. This is a keyboard Local Navigation command.

Default Values: n+Ctrl, P+Ctrl, Down, R14, Down+Shift
Values (Basic or Full): Down

OpenWindows.KeyboardCommand.Empty

This mapping is always loaded. This is a Text Editing command resource.

Values: e+Meta, e+Ctrl+Meta

OpenWindows.KeyboardCommand.EraseCharBackward

This mapping is always loaded. This is a Text Editing command resource.

Values: Delete, BackSpace

OpenWindows.KeyboardCommand.EraseCharForward

This mapping is always loaded. This is a Text Editing resource.

Values: Delete+Shift, BackSpace+Shift

OpenWindows.KeyboardCommand.EraseLine

This is a Text Editing resource.

Values (Basic or Full): Delete+Meta, BackSpace+Meta

OpenWindows.KeyboardCommand.EraseLineBackward

This mapping is always loaded. This is a Text Editing resource.

Values: u+Ctrl

OpenWindows.KeyboardCommand.EraseLineEnd

This mapping is always loaded. This is a Text Editing resource.

Values: U+Ctrl

OpenWindows.KeyboardCommand.EraseWordBackward

This mapping is always loaded. This is a Text Editing resource.

Values: w+Ctrl

OpenWindows.KeyboardCommand.EraseWordForward

This mapping is always loaded. This is a Text Editing resource.

Values: W+Ctrl

OpenWindows.KeyboardCommand.FindBackward

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: F+Meta, L9+Shift

OpenWindows.KeyboardCommand.FindForward

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: f+Meta, L9

OpenWindows.KeyboardCommand.GoLineForward

This mapping is always loaded. This is a keyboard Local Navigation command.

Values: apostrophe+Ctrl, R11

OpenWindows.KeyboardCommand.GoPageBackward

This mapping is always loaded. This is a keyboard Local Navigation command.

Values: R9

OpenWindows.KeyboardCommand.GoPageForward

This mapping is always loaded. This is a keyboard Local Navigation command.

Values: R15

OpenWindows.KeyboardCommand.GoWordForward

This mapping is always loaded. This is a keyboard Local Navigation command.

Values: slash+Ctrl, less+Ctrl

OpenWindows.KeyboardCommand.Help

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: Help

OpenWindows.KeyboardCommand.HorizontalScrollbarMenu

This is a Full “Miscellaneous Navigation” command.

Values: h+Alt

OpenWindows.KeyboardCommand.IncludeFile

This mapping is always loaded. This is a Text Editing resource.

Values: i+Meta

OpenWindows.KeyboardCommand.InputFocusHelp

This mapping is loaded if KeyboardCommand is Full. This is a keyboard Core Functions resource.

Values:	question+Ctrl

OpenWindows.KeyboardCommand.Insert

This mapping is always loaded. This is a Text Editing resource.

Values:	Insert

OpenWindows.KeyboardCommand.JumpDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values:	Down+Ctrl

OpenWindows.KeyboardCommand.JumpLeft

This mapping is always loaded. This is a keyboard Local Navigation command.

Default Values:	comma+Ctrl, greater+Ctrl
Values (Basic or Full):	Left+Ctrl

OpenWindows.KeyboardCommand.JumpMouseToInputFocus

This mapping is loaded if KeyboardCommand is Full. This is a keyboard Core Functions resource.

Values:	j+Alt

OpenWindows.KeyboardCommand.JumpRight

This mapping is always loaded. This is a keyboard Local Navigation command.

Default Values:	period+Ctrl
Values (Basic or Full):	Right+Ctrl

OpenWindows.KeyboardCommand.JumpUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values:	Up+Ctrl

OpenWindows.KeyboardCommand.Left

This mapping is always loaded. This is a keyboard Local Navigation command.

Values:	b+Ctrl, F+Ctrl, Left, R10, Left+Shift
Values (Basic or Full):	Left

OpenWindows.KeyboardCommand.LineEnd

This mapping is always loaded. This is a keyboard Local Navigation command.

Values:	e+Ctrl, A+Ctrl

OpenWindows.KeyboardCommand.LineStart

This mapping is always loaded. This is a keyboard Local Navigation command.

Values: a+Ctrl, E+Ctrl

OpenWindows.KeyboardCommand.Load

This mapping is always loaded. This is a Text Editing resource.

Values: l+Meta

OpenWindows.KeyboardCommand.MatchDelimiter

This mapping is always loaded. This is a Text Editing resource.

Values: d+Meta

OpenWindows.KeyboardCommand.Menu

This mapping is loaded if KeyboardCommand is Full. This is a keyboard Core Functions resource.

Values: space+Alt

OpenWindows.KeyboardCommand.MoreHelp

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: Help+Shift

OpenWindows.KeyboardCommand.MoreTextHelp

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: Help+Shift+Ctrl

OpenWindows.KeyboardCommand.NextElement

This mapping is loaded if KeyboardCommand is Full. This is a Global Navigation command.

Values: Tab+Ctrl

OpenWindows.KeyboardCommand.NextPane

This mapping is loaded if KeyboardCommand is Full. This is a Global Navigation command.

Values: a+Alt

OpenWindows.KeyboardCommand.PaneBackground

This mapping is loaded if KeyboardCommand is Full. This is a Miscellaneous Navigation command.

Values: b+Alt

OpenWindows.KeyboardCommand.PaneDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values: R15

OpenWindows.KeyboardCommand.PaneLeft

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values: R9+Ctrl

OpenWindows.KeyboardCommand.PaneRight

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values: R15+Ctrl

OpenWindows.KeyboardCommand.PaneUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values: R9

OpenWindows.KeyboardCommand.PanelEnd

This mapping is loaded if KeyboardCommand is Full. This is a Miscellaneous Navigation command.

Values: bracketright+Ctrl

OpenWindows.KeyboardCommand.PanelStart

This mapping is loaded if KeyboardCommand is Full. This is a Miscellaneous Navigation command.

Values: bracketleft+Ctrl

OpenWindows.KeyboardCommand.Paste

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: v+Meta, L8

OpenWindows.KeyboardCommand.PreviousElement

This mapping is loaded if KeyboardCommand is Full. This is a Global Navigation command.

Values: Tab+Shift+Ctrl

OpenWindows.KeyboardCommand.PreviousPane

This mapping is loaded if KeyboardCommand is Full. This is a Global Navigation command.

Values: A+Alt

OpenWindows.KeyboardCommand.Props

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: L3

OpenWindows.KeyboardCommand.QuoteNextKey

This mapping is loaded if KeyboardCommand is Full. This is a keyboard Core Functions resource.

Values: q+Alt

OpenWindows.KeyboardCommand.ResumeMouseless

This mapping is loaded if KeyboardCommand is Full. This is a keyboard Core Functions resource.

Values: Z+Alt

OpenWindows.KeyboardCommand.Right

This mapping is always loaded. This is a keyboard Local Navigation command.

Default Values: f+Ctrl, B+Ctrl, Right, R12, Right+Shift
Values (Basic or Full): Right

OpenWindows.KeyboardCommand.RowEnd

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values: End, R13

OpenWindows.KeyboardCommand.RowStart

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Local Navigation command.

Values: Home, R7

OpenWindows.KeyboardCommand.ScrollDataEnd

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: End+Alt+Ctrl, R13+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollDataStart

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Home+Alt+Ctrl, R7+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Down+Alt

OpenWindows.KeyboardCommand.ScrollJumpDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Down+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollJumpLeft

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Left+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollJumpRight

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Right+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollJumpUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Up+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollLeft

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Left+Alt

OpenWindows.KeyboardCommand.ScrollPaneDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R15+Alt

OpenWindows.KeyboardCommand.ScrollPaneLeft

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R9+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollPaneRight

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R15+Alt+Ctrl

OpenWindows.KeyboardCommand.ScrollPaneUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R9+Alt

OpenWindows.KeyboardCommand.ScrollRight

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Right+Alt

OpenWindows.KeyboardCommand.ScrollRowEnd

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: End+Alt, R13+Alt

OpenWindows.KeyboardCommand.ScrollRowStart

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Home+Alt, R7+Alt

OpenWindows.KeyboardCommand.ScrollUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Up+Alt

OpenWindows.KeyboardCommand.SelectAll

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: End+Shift+Meta

OpenWindows.KeyboardCommand.SelectDataEnd

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: End+Shift+Ctrl, R13+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectDataStart

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Home+Shift+Ctrl, R7+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Down+Shift

OpenWindows.KeyboardCommand.SelectFieldBackward

This mapping is always loaded. This is a Text Editing resource.

Values: Tab+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectFieldForward

This mapping is always loaded. This is a Text Editing resource.

Values: Tab+Ctrl

OpenWindows.KeyboardCommand.SelectJumpDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Down+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectJumpLeft

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Left+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectJumpRight

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Right+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectJumpUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Up+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectLeft

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Left+Shift

OpenWindows.KeyboardCommand.SelectNextField

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Tab+Meta

OpenWindows.KeyboardCommand.SelectPaneDown

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R15+Shift

OpenWindows.KeyboardCommand.SelectPaneLeft

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R9+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectPaneRight

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R15+Shift+Ctrl

OpenWindows.KeyboardCommand.SelectPaneUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: R9+Shift

OpenWindows.KeyboardCommand.SelectPreviousField

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Tab+Shift+Meta

OpenWindows.KeyboardCommand.SelectRight

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Right+Shift

OpenWindows.KeyboardCommand.SelectRowEnd

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: End+Shift, R13+Shift

OpenWindows.KeyboardCommand.SelectRowStart

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Home+Shift, R7+Shift

OpenWindows.KeyboardCommand.SelectUp

This mapping is loaded if KeyboardCommand is Basic or Full. This is a Text Editing resource.

Values: Up+Shift

OpenWindows.KeyboardCommand.Stop

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: L1

OpenWindows.KeyboardCommand.Store

This mapping is always loaded.

Values: s+Meta

OpenWindows.KeyboardCommand.SuspendMouseless

This mapping is loaded if KeyboardCommand is Full. This is a keyboard Core Functions resource.

Values: z+Alt

OpenWindows.KeyboardCommand.TextHelp

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: Help+Ctrl

OpenWindows.KeyboardCommand.Translate

This mapping is loaded if KeyboardCommand is Basic or Full. This is a keyboard Core Functions resource.

Values: R2

OpenWindows.KeyboardCommand.Undo

This mapping is always loaded. This is a keyboard Core Functions resource.

Values: u+Meta, L4

OpenWindows.KeyboardCommand.Up

This mapping is always loaded. This is a keyboard Local Navigation command.

Default Values: p+Ctrl, N+Ctrl, Up, R8, Up+Shift
Values (Basic or Full): Up

OpenWindows.KeyboardCommand.VerticalScrollbarMenu

This mapping is loaded if KeyboardCommand is Full. This is a Miscellaneous Navigation command.

Values: v+Alt

OpenWindows.KeyboardCommands

All of the `OpenWindows.KeyboardCommand` resource mappings may be modified by users, or by specifying one of three values for `OpenWindows.KeyboardCommands`. See the description for the Mouseless model in Chapter 6, *Handling Input*, in the *XView Programming Manual*.

Values: Sunview1, Basic, or Full.

OpenWindows.MonospaceFont

Compatibility is maintained with the older resources `Font.Name` and `Window.Scale`. This sets the font used by cmdtool, shelltool and any text subwindow or text editor that uses a fixed width font. If *-font* is specified on the command line, then it will override the font set by `OpenWindows.MonospaceFont`. To find out what fonts are available, use the *xlsfonts* command. This is a string. The default is `Null`.

Values: font name

OpenWindows.MouseChordMenu

Turns on the mouse chording mechanism. Mouse chording allows XView to work with two button mice. Holding the SELECT and the ADJUST buttons together will act as MENU button.

Values: Boolean (False)

OpenWindows.MouseChordTimeout

Mouse chording time-out value.

Values: Integer in microseconds (100)

OpenWindows.multiClickTimeout (Props)

Specifies a “click”, which is button-down, button-up pair.

Values: N (4)
Where N is an integer greater than 2. Set the number of tenths of a second between clicks for a multi-click.

OpenWindows.popupJumpCursor (Props)

When False, do not warp the mouse to the notice when it appears.

Values: True, False (False)

OpenWindows.RegularFont

Sets the font used by panels, menus and notices. If *-font* is specified on the command line, then it will override the font set by `Openwindows.RegularFont`. To find out what fonts are available, use the *xlsfonts* command. The Default is `Null`.

Values: font name

OpenWindows.Scale

Refer to the description for -scale.

Values: Small, Medium, Large, Extra-Large

OpenWindows.scrollbarPlacement (Props)

When set to Left, put all scrollbars on the left-hand side of the window or object.

Values: Left, Right (Right)

OpenWindows.SelectDisplaysMenu (Props)

When True, the SELECT button (usually left mouse) will display the menu as well as the MENU button (usually right mouse).

Values: True, False (False)

OpenWindows.windowColor (Props)

Specifies the base color for control areas for a 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.

Values: Any valid X11 color specification (#cccccc — 80% gray)

OpenWindows.workspaceColor (Props)

Specifies the color for the root window and the background color for icons that blend into the desktop.

Values: Any valid X11 color specification (#cccccc — 80% gray)

scrollbar.jumpCursor (Props)

When False, the scrollbar will not move the mouse pointer when scrolling.

Values: True, False (True)

scrollbar.lineInterval

Specifies the time in milliseconds between repeats of a single line scroll. This indicates how long to pause scrolling when holding down the SELECT button on the scrollbar elevator. Scrollbar sets up a timer routine for repeats.

Values: N (1)
Where N is some integer greater than 0.

scrollbar.pageInterval

Specifies the time in milliseconds between repeats of a single page scroll.

Values: N (100)
Where N is some integer greater than 2.

scrollbar.repeatDelay

Specifies the time in milliseconds when a click becomes a repeated action.

Values: N (100)
Where N is some integer greater than 2.

Selection.Timeout

Selection timeout value. This value indicates the number of seconds that a requestor or a selection owner waits for a response.

Values: Integer (3)

term.alternateTtyswrc

This is only used if a *.ttyswrc* file is not found in *$HOME/.ttyswrc* and `term.useAlternateTtyswrc` is True.

Values: filename (*$XVIEWHOME/lib/.ttyswrc*)
Where filename specifies a complete filename and absolute path of an alternate *.ttyswrc* file.

term.boldStyle

Specifies the text bolding style for a terminal-based window.

Values: None, Offset_X, Offset_Y, Offset_X_and_Y_and_XY, Offset_XY, Offset_X_and_XY, Offset_Y_and_XY, Offset_X_and_Y, Invert (Invert)

term.enableEdit

When False, do not keep an edit log 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.

Values: True, False (True)

term.inverseStyle

Specifies the text inverting style for a terminal-based window.

Values: Enable, Disable, Same_as_bold (Enable)

term.underlineStyle

Specifies the text underlining style for a terminal-based window.

Values: Enable, Disable, Same_as_bold (Enable)

term.useAlternateTtyswrc

When True, and a *$HOME/.ttyswrc* is not found, look for an alternate *.ttyswrc* file. When False, do not look for an alternate file if one is not found in the home directory, *$HOME/.ttyswrc*.

Values: True, False (True)

text.againLimit

Number of operations the "again history" remembers for a textsw.

Values: N (1)
Where N is an integer between 0 and 500.

text.autoIndent

When True, begin the next line at the same indentation as the previous line as typing in text.

Values: True, False (False)

text.autoScrollBy

Specifies the number of lines to scroll when type-in moves insertion point below the view.

Values: N (1)
Where N is an integer between 0 and 100.

text.blinkCaret

Values: True, False (True)

text.confirmOverwrite

When False, do not give user confirmation if a save will overwrite an existing file.

Values: True, False (True)

Text.DeleteReplacesClipboard

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 will not be copied to the clipboard. The default value is False, do not replace the contents of 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. The default value is False, do not replace the contents of the clipboard. (in XView V3.0.1, the value for this resource was effectively True, the text selected always replaced the clipboard.)

Values: True, False (False)

text.delimiterChars

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.

The selection of delimiters will be automatically available to the user once the SunOS 4.1 becomes the default operating system environment, however this resource is used as a bridge during that period.

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 string. The specific problem(s) that occur are the stripping of the backslash characters and the expansion of the tab character (11).

To correct this problem, one can put the `text.delimiterChars` 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.delimiterChars` entry can be inserted after *.Xdefaults* file saves.

Values: string
The default follows:

```
\t,.:;?!’"‘*/–+=(){}[ ]<>\|˜@#$%ˆ&
```

text.displayControlChars

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.

Values: True, False (True)

text.enableScrollbar

When False, do not put a scrollbar on the text window.

Values: True, False (True)

text.extrasMenuFilename

The file specified by this resource is used for the text package’s Extras menu. This resource can be specified with or without `.<`*`locale`*`>` appended to the resource name. `text.extrasMenuFilename.<`*`locale`*`>` will take precedence over `text.extrasMenuFilename`, where `<`*`locale`*`>` is the current locale value of the display lang category (`XV_LC_DISPLAY/LC_MESSAGES`). A command executed in the extras menu is applied to the contents of the current selection in the textsw window. The results are inserted at the current insertion point.

Values: Filename (*/usr/lib/.text_extras_menu*) where filename is an absolute pathname of a file.

Default:	This value may also be set via environment variables EXTRASMENU.<*locale*> or EXTRASMENU. If neither resource or environment variable exist, the text package will try to locate an Extras menu in one of the following: *$(HOME)/.text_extras_menu[.<locale>]* *$(OPENWINHOME)/lib/locale/<locale>/xview/.text_extras_menu* *$(OPENWINHOME)/lib/locale/C/xview/.text_extras_menu* */usr/lib/.text_extras_menu*.

Text.FilterReplacesClipboard.

If the value is True, then the text selected for a filter command (e.g. the Extras->Format menu item from the textsw menu) will replace the contents of the clipboard after the filter has successfully completed. If the value is False, then the text selected will not replace the contents of the clipboard. The default value is False, do not replace the contents of the clipboard.

Values: True, False (False)

text.insertMakesCaretVisible

Controls whether insertion causes repositioning to make inserted text visible.

Values: `If_auto_scroll` (Always)

text.lineBreak

Determines how the textsw treats file lines when they are too big to fit on one display line.

Values: Clip, Wrap_char, Wrap_word (Wrap_word)

text.lineSpacing

This resource takes an integer value which is the percentage of the maximum height of a character in the textsw window font to use as interline spacing. For example, the maximum character height in the textsw window font is 10. Setting `Text.LineSpacing` to 15 will cause 2 pixels (1.5 pixels rounded up) of white space to appear between each line in the textsw.

Values: integer (0)

text.margin.bottom

Specifies the minimum number of lines to maintain between insertion point and bottom of view. A value of –1 turns auto scrolling off.

Values: N (0)
Where N is an integer between –1 and 50.

text.margin.left

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.

Values: N (8)
Where N is an integer between 0 and 2000.

text.margin.right

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.

Values:	N (0) Where N is an integer between 0 and 2000.

text.margin.top

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.

Values:	N (2) Where N is an integer between –1 and 50.

text.maxDocumentSize

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.

Values:	N (20000) Where N specifies the bytes used in memory before a text file is saved to a file on disk.

text.retained

If True, retain text windows with server backing store.

Values:	True, False (False)

text.storeChangesFile

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.

Values:	True, False (True)

text.tabWidth

Specifies the width in characters of the tab character.

Values:	N (8) Where N is an integer between 0 and 50.

text.undoLimit

Specifies 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.

Values:	N (50 maximum of 500) Where N is an integer between 0 and 500.

tty.eightBitOutput

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.

Values:	Boolean Default is True.

tty.yieldModifiers

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 a termsw would be removed. The default is to not remove any semantic meaning from any modifiers.

Values:	List of modifier keys Alt and Meta are the only two keys currently supported.

window.synchronous, +sync -sync

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.

Values:	True, False (False)

Window.Depth

Specifies the window's depth. Use an integer.

Window.visual

Resource specifies the visual used for the window.

xview.icccmcompliant

When False, XView will set window manager hints in a way that was used before the ICCCM was adopted. Useful for window managers that are released before X11R4. Not needed with the Open Look Window Manager provided with Open Windows.

Values:	True, False (True)