$RCSfile: header.man,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:20 $ # $RCSfile: ButtonList.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:23 $

NAME

ButtonList — List of available Buttons for Automation

DESCRIPTION

The **ButtonList** is a List of the available Buttons to be pressed, released, and clicked within Automation scripts. This is derived from the *X Window System Specifications*.

Btn1 Mouse Button1.

Btn2 Mouse Button2.

Btn3 Mouse Button3.

Btn4 Mouse Button4.

Btn5 Mouse Button5.

For more information about **ButtonList**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

RELATED INFORMATION

PressMB(3X), **ReleaseMB(3X)**, **ClickMB(3X)**, **DragComponent(3X)**, **DragXYRelative(3X)**, **DragSliderValue(3X)**, **DragXYAbs**, **ModifierList(3X)** # $RCSfile: ClickKey.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:24 $

NAME

ClickKey — Convenience routine to press and release a key.

SYNOPSIS

ClickKey *[ModifierList]* Key*

DESCRIPTION

The **ClickKey** command will simulate the clicking of any key on the standard keyboard, including Motif Virtual Keys. The key click maybe modified by any combination of the standard X modifiers. This function usually follows a command to **LocatePointer** to a widget which accepts keyboard input. The key click is composed of a key press immediately followed by a key release.

*[ModifierList]**
: List of Modifiers to modify the key click. For a full list of available Modifiers see **ModifierList(3X)**. Default is no modifiers.

Key
: Key to be clicked. For a full list of available Keys see **KeyList(3X)**.

EXAMPLE

ClickKey <Shift> KPageDown

This will click the KPageDown key with the modifier Shift.

For more information about **ClickKey**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

RELATED INFORMATION

LocatePointer(3X), **PressKey(3X)**, **ReleaseKey(3X)**, **KeyList(3X)**, **ModifierList(3X)** # $RCSfile: ClickMB.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:25 $

NAME

ClickMB — Command to click any mouse button.

SYNOPSIS

ClickMB *[ModifierList]* Button [Count]*

DESCRIPTION

The **ClickMB** command will simulate the clicking of a button on the standard mouse. The button click may be modified by any combination of standard X modifiers. This function is usually follows a command to **LocatePointer** to a widget which accepts mouse input. The mouse button click consists of a mouse button press immediately followed by a mouse button release. This command takes an optional integer argument *count* that allows the synthesis of multiple clicks.

*[ModifierList]**
: List of Modifiers to modify the mouse button click. For a full list of available Modifiers see **ModifierList(3X)**. Default is no modifiers.

Button
: Button to be clicked. For a full list of available Buttons see **ButtonList(3X)**.

[Count]
: The multiclick count. Default count is 1.

For more information about **ClickMB**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

ClickMB <Shift> <Ctrl> Btn1 2

This will double click mouse Button1 with Control and Shift modifiers.

RELATED INFORMATION

LocatePointer(3X), **PressMB(3X)**, **ReleaseMB(3X)**, **ButtonList(3X)**, **ModifierList(3X)** # $RCSfile: CompareVisual.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:26 $

NAME

CompareVisual — Command to compare/record widget visuals.

SYNOPSIS

CompareVisual *WidgetName [Component]*

DESCRIPTION

The **CompareVisual** command will either record or compare the visual of the widget *WidgetName*. In record mode, **CompareVisual** will record a visual image of the specified widget. In playback mode, **CompareVisual** will compare the requested widget or component of the widget against the image stored in the visual comparison file. If the stored image cannot be read, a warning will be issued and visual comparisons will be turned off for the remainder of the test. The visual algorithm records visual images independently of the current colormap, therefore the specific colors used will not be compared. If the environment variable **$AUTOVPATH** is set, this path will be prepended to the name of the visual comparison file for writing or reading the visual file.

WidgetName
: The XtName representing the widget you wish to take an action on.

[Component]
: Particular component within the widget. See **ComponentList(3X)** for a full list of supported Components.

For more information about **CompareVisual**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

CompareVisual ArrowButton1

This will either record or compare the widget ArrowButton1. If the test is run with the -r option visuals will be recorded, else the visual comparisons will take place.

RELATED INFORMATION

LocatePointer(3X) # $RCSfile: ComponentList.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:27 $

NAME

ComponentList — List of available Components for Automation

DESCRIPTION

The **ComponentList** is a List of the available component regions for use with Automation. A component describes a particular region of the widget in question. A component is usually another widget inside a composite widget. An example of a component is the Help Button in a MessageBox. This component is actually a Push Button, but we need know only its component name, not its actual **XtName, to reference it. The following is a list of valid components for all Motif widgets. There is some overlap in the components. All component names start with a capital R.**

XmCommand:

RSelectionList
Describes the Command's List widget.

RSelectionListItem
Describes a List item within the Command's List widget.

RSelectionText
Describes the Command's Text widget.

RSelectionTextChar
Describes a Text character within the Command's Text widget.

RLabel Describes the Label above the Command's Text widget.

RSelectListHorizScrollBarLeftArrow
Describes the horizontal ScrollBar's left arrow (if it exists).

RSelectListHorizScrollBarRightArrow
Describes the horizontal ScrollBar's right arrow (if it exists).

RSelectListHorizScrollBarLeftSliderArea
Describes the area in the horizontal ScrollBar's trough to the left of the current slider position (if it exists).

RSelectListHorizScrollBarRightSliderArea
Describes the area in the horizontal ScrollBar's trough to the right of the current slider position (if it exists).

RSelectListHorizScrollBarSlider
Describes the midpoint of the horizontal ScrollBar's slider (if it exists).

RSelectListVertScrollBarUpArrow
Describes the vertical ScrollBar's up arrow (if it exists).

RSelectListVertScrollBarDownArrow
Describes the vertical ScrollBar's down arrow (if it exists).

RSelectListVertScrollBarLeftSliderArea
Describes the area in the vertical ScrollBar's trough to the left of the current slider position (if it exists).

RSelectListVertScrollBarRightSliderArea
Describes the area in the vertical ScrollBar's trough to the right of the current slider position (if it exists).

RSelectListVertScrollBarSlider
Describes the midpoint of the vertical ScrollBar's slider (if it exists).

XmFileSelectionBox:

RFileSelectionText
Describes the text region for displaying and editing a filename.

RFileSelectionTextChar
Describes a character within the selection Text Widget. Characters are numbered starting at 0.

RFileSelectionLabel
Describes the label above the selection Text Widget.

RFileFilterText
Describes the text region for displaying and editing a directory mask used to select the files to be displayed.

RFileFilterTextChar
Describes a character within the filter Text Widget. Characters are numbered starting at 0.

RFileFilterLabel
Describes the label above the filter Text Widget.

RFileList
Describes the scrollable List which contains filenames.

RFileListLabel
Describes the label above the filenames List.

RFileListItem
Describes a List item in the filenames List. 0 defines the first item position in the List.

RDirList
Describes the scrollable List which contains subdirectories.

RDirListLabel
Describes the label above the directory List.

RDirListItem
Describes a List item in the directory List. 0 defines the first item position in the List.

RSeparator
Describes the Separator widget.

ROkButton
Describes the FileSelectionBox's OK PushButton.

RCancelButton
Describes the FileSelectionBox's Cancel PushButton.

RFilterButton
Describes the FileSelectionBox's Filter (Apply) PushButton.

RHelpButton
Describes the FileSelectionBox's Help PushButton.

RFileListHorizScrollBarLeftArrow
Describes the File List's horizontal ScrollBar's left arrow.

RFileListHorizScrollBarRightArrow
Describes the File List's horizontal ScrollBar's right arrow.

RFileListHorizScrollBarLeftSliderArea
Describes the area in the File List's horizontal ScrollBar's trough to the left of the current slider position.

RFileListHorizScrollBarRightSliderArea
Describes the area in the File List's horizontal ScrollBar's trough to the right of the current slider position.

RFileListHorizScrollBarSlider
Describes the midpoint of the File List's horizontal ScrollBar's slider.

RFileListVertScrollBarUpArrow
Describes the File List's vertical ScrollBar's up arrow.

RFileListVertScrollBarDownArrow
Describes the File List's vertical ScrollBar's down arrow.

RFileListVertScrollBarLeftSliderArea
Describes the area in the File List's vertical ScrollBar's trough to the left of the current slider position.

RFileListVertScrollBarRightSliderArea
Describes the area in the File List's vertical ScrollBar's trough to the right of the current slider position.

RFileListVertScrollBarSlider
Describes the midpoint of the File List's vertical ScrollBar's slider.

RDirListHorizScrollBarLeftArrow
Describes the Dir List's horizontal ScrollBar's left arrow.

RDirListHorizScrollBarRightArrow
Describes the Dir List's horizontal ScrollBar's right arrow.

RDirListHorizScrollBarLeftSliderArea
Describes the area in the Dir List's horizontal ScrollBar's trough to the left of the current slider position.

RDirListHorizScrollBarRightSliderArea
Describes the area in the Dir List's horizontal ScrollBar's trough to the right of the current slider position.

RDirListHorizScrollBarSlider
Describes the midpoint of the Dir List's horizontal ScrollBar's slider.

RDirListVertScrollBarUpArrow
Describes the Dir List's vertical ScrollBar's up arrow.

RDirListVertScrollBarDownArrow
Describes the Dir List's vertical ScrollBar's down arrow.

RDirListVertScrollBarLeftSliderArea
Describes the area in the Dir List's vertical ScrollBar's trough to the left of the current slider position.

RDirListVertScrollBarRightSliderArea
Describes the area in the Dir List's vertical ScrollBar's trough to the right of the current slider position.

RDirListVertScrollBarSlider
Describes the midpoint of the Dir List's vertical ScrollBar's slider.

XmList:

RListItem
Describes an item contained in a List. 0 defines the first item position in the List.

RHorizScrollBarLeftArrow
Describes the horizontal ScrollBar's left arrow (if it exists).

RHorizScrollBarRightArrow
Describes the horizontal ScrollBar's right arrow (if it exists).

RHorizScrollBarLeftSliderArea
Describes the area in the horizontal ScrollBar's trough to the left of the current slider position (if it exists).

RHorizScrollBarRightSliderArea
Describes the area in the horizontal ScrollBar's trough to the right of the current slider position (if it exists).

RHorizScrollBarSlider
Describes the midpoint of the horizontal ScrollBar's slider (if it exists).

RVertScrollBarUpArrow
Describes the vertical ScrollBar's up arrow (if it exists).

RVertScrollBarDownArrow
Describes the vertical ScrollBar's down arrow (if it exists).

RVertScrollBarUpSliderArea
Describes the area in the vertical ScrollBar's trough above the slider of the current slider position (if it exists).

RVertScrollBarDownSliderArea
Describes the area in the vertical ScrollBar's trough below the slider of the current slider position (if it exists).

RVertScrollBarSlider
Describes the midpoint of the vertical ScrollBar's slider (if it exists).

XmMainWindow:

RCommandWindow
Describes the command widget associated with the MainWindow.

RMessageWindow
Describes the message widget associated with the MainWindow.

RMenuBar
Describes the MenuBar widget associated with the MainWindow.

RWorkWindow
Describes the work area widget associated with the MainWindow.

RSeparator
Describes the Separator widgets associated with the MainWindow.

RVertScrollBar
Describes the vertical ScrollBar widget associated with the MainWindow.

RHorizScrollBar
Describes the horizontal ScrollBar widget associated with the MainWindow.

XmMessageBox:

ROkButton
Describes the MessageBox's OK button.

RCancelButton
Describes the MessageBox's Cancel button.

RHelpButton
Describes the MessageBox's Help button.

RMessageLabel
Describes the MessageBox's Message label string.

RSeparator
Describes the MessageBox's Separator.

RSymbolLabel
Describes the MessageBox's message symbol.

XmOptionMenu:

ROptionButton
Describes the Option menu's selection area.

RLabel Describes the Option menu's Label.

XmPanedWindow:

RSash Describes the sashes contained in the PanedWindow. Sash 0 is the poition of the first sash in the PanedWindow.

XmPulldown:

RTearOffButton
Describes the TearOffButton associated with the Pulldown menu (If it exists).

XmPopupMenu:

RTearOffButton
Describes the TearOffButton associated with the Popup menu (If it exists).

XmScale:

RScaleLabel
Describes the Label associated with the Scale.

RScaleUpOrLeftSliderArea
Describes the region of the Scale to the left or above the slider in the trough (depending on the Scale's orientation).

RScaleDownOrRightSliderArea.
Describes the region of the Scale to the right or below the slider in the trough (depending on the Scale's orientation).

RScaleSlider
Describes the midpoint of the Scale's slider.

XmScrollBar:

RUpOrLeftSliderArea
Describes the region of the Scale to the left or above the slider in the trough (depending on the ScrollBars's orientation).

RDownOrRightSliderArea
Describes the region of the Scale to the right or below the slider in the trough (depending on the ScrollBars's orientation).

RSlider
Describes the midpoint of the Scale's slider.

RDownOrRightArrow
Describes the Top or Right arrow in ScrollBar (depending on the ScrollBars's orientation).

RUpOrLeftArrow
Describes the Bottom or Left arrow in ScrollBar (depending on the ScrollBars's orientation).

XmScrolledWindow:

RHorizScrollBarLeftArrow
Describes the horizontal ScrollBar's left arrow (if it exists).

RHorizScrollBarRightArrow
Describes the horizontal ScrollBar's right arrow (if it exists).

RHorizScrollBarLeftSliderArea
: Describes the area in the horizontal ScrollBar's trough to the left of the current slider position (if it exists).

RHorizScrollBarRightSliderArea
: Describes the area in the horizontal ScrollBar's trough to the right of the current slider position (if it exists).

RHorizScrollBarSlider
: Describes the midpoint of the horizontal ScrollBar's slider (if it exists).

RVertScrollBarUpArrow
: Describes the vertical ScrollBar's up arrow (if it exists).

RVertScrollBarDownArrow
: Describes the vertical ScrollBar's down arrow (if it exists).

RVertScrollBarDownSliderArea
: Describes the area in the vertical ScrollBar's trough to the below of the current slider position (if it exists).

RVertScrollBarUpSliderArea
: Describes the area in the vertical ScrollBar's trough to the above of the current slider position (if it exists).

RVertScrollBarSlider
: Describes the midpoint of the vertical ScrollBar's slider (if it exists).

XmSelectionBox:

RSelectionList
: Describes the SelectionBox's ScrolledList of alternatives.

RSelectionListItem
: Describes a List item in the selection List. 0 defines the first item position in the List.

RSelectListHorizScrollBarLeftArrow
: Describes the SelectionBox List's horizontal Scrollbar left arrow (if it exists).

RSelectListHorizScrollBarRightArrow
: Describes the SelectionBox List's horizontal Scrollbar right arrow (if it exists).

RSelectListHorizScrollBarLeftSliderArea
: Describes the area in the SelectionBox List's horizontal ScrollBar's trough to the left of the current slider position.

RSelectListHorizScrollBarRightSliderArea
: Describes the area in the SelectionBox List's horizontal ScrollBar's trough to the right of the current slider position.

RSelectListHorizScrollBarSlider
: Describes the midpoint of the SelectionBox List's horizontal ScrollBar's slider.

RSelectListVertScrollBarUpArrow
: Describes the SelectionBox List's vertical Scrollbar up arrow (if it exists).

RSelectListVertScrollBarDownArrow
: Describes the SelectionBox List's vertical Scrollbar down arrow (if it exists).

RSelectListVertScrollBarUpSliderArea
: Describes the area in the SelectionBox List's vertical ScrollBar's trough above the current slider position.

RSelectListVertScrollBarDownSliderArea
: Describes the area in the SelectionBox List's vertical ScrollBar's trough below the

current slider position.

RSelectListVertScrollBarSlider
Describes the midpoint of the SelectionBox List's vertical ScrollBar's slider.

RSelectionListLabel
Describes the Label associated with the SelectionBox's List.

RSelectionText
Describes the SelectionBox's Text widget for the selected alternative.

RSelectionTextChar
Describes a character in the SelectionBox's Text widget.

RSelectionLabel
Describes the label for the SelctionBox's Text widget.

ROkButton
Describes the SelectionBox's OK PushButton.

RCancelButton
Describes the SelectionBox's Cancel PushButton.

RHelpButton
Describes the SelectionBox's Help PushButton.

RApplyButton
Describes the SelectionBox's Apply PushButton.

RSeparator
Describes the SelectionBox's Separators

XmText:

RTextChar
Describes a single character in the Text widget. Characters are numbered starting at 0.

RHorizScrollBarLeftArrow
Describes the horizontal ScrollBar's left arrow (if it exists).

RHorizScrollBarRightArrow
Describes the horizontal ScrollBar's right arrow (if it exists).

RHorizScrollBarLeftSliderArea
Describes the area in the horizontal ScrollBar's trough to the left of the current slider position (if it exists).

RHorizScrollBarRightSliderArea
Describes the area in the horizontal ScrollBar's trough to the right of the current slider position (if it exists).

RHorizScrollBarSlider
Describes the midpoint of the horizontal ScrollBar's slider (if it exists).

RVertScrollBarUpArrow
Describes the vertical ScrollBar's up arrow (if it exists).

RVertScrollBarDownArrow
Describes the vertical ScrollBar's down arrow (if it exists).

RVertScrollBarLeftSliderArea
Describes the area in the vertical ScrollBar's trough to the left of the current slider position (if it exists).

RVertScrollBarRightSliderArea
Describes the area in the vertical ScrollBar's trough to the right of the current slider

position (if it exists).

RVertScrollBarSlider
Describes the midpoint of the vertical ScrollBar's slider (if it exists).

For more information about **ButtonList**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

RELATED INFORMATION

LocatePointer(3X), **DragComponent(3X)** # $RCSfile: Continue.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:28 $

NAME

Continue — Command to continue an Automated script

SYNOPSIS

Continue

DESCRIPTION

The **Continue** command will continue the current test being executed. First, the code performs a **LocatePointer** to the OK (**Continue**) button of the instruction box, and then and synthesizes a button press on Button1. Each **Continue** in the command script should have a corresponding CommonPause() in the client test program. The program execute all scripting instructions up to the **Continue** command before continuing with the next set of commands.

For more information about **Continue**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

Continue

This will continue the current test.

RELATED INFORMATION

Exit(3X) # $RCSfile: DragComponent.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:29 $

NAME

DragComponent — Command to drag within a component of a widget.

SYNOPSIS

DragComponent *WidgetName Component Destination [ModifierList]* [Button]*

DESCRIPTION

The **DragComponent** command will drag between two subcomponents of a widget. This command will be used in a List widget to drag between List items, and also will be used in a Text widget to drag between text characters (to make primary and secondary selections). This command should follow a command to **LocatePointer** to a subcomponent where the drag will begin. The **DragComponent** command will then cause a drag action between the two points, current and specified. The **DragComponent** command is composed of a call to **PressMB**, a call to **LocatePointer** to move to the destination, and then a call to **ReleaseMB**. The mouse button used in the drag may be specified with optional argument *Button*.

WidgetName The XtName representing the widget you wish to take an action on.

Component Particular component within the widget. See **ComponentList(3X)** for a full list of supported components.

Destination Number correponding to the index within the component. In the case where the component is *RListItem*, *Destination* refers to the list item number (0 is the first element in the list). If the component is *RTextChar*, *Destination* refers to the character position with the Text widget (0 is the first character).

*[ModifierList]**
List of Modifiers to modify the mouse button press. For a full list of available Modifiers see **ModifierList(3X)**. The default is no modifiers.

[Button] Button to be pressed. For a full list of available Buttons see **ButtonList**. The default is **Button1**.

For more information about **DragComponent**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

LocatePointer Text1 RTextChar 25
DragComponent Text1 RTextChar 30 <Shift> Btn2

These two commands will drag between character 25 and character 30 with Button2 pressed, modified by Shift.

RELATED INFORMATION

LocatePointer(3X), **PressMB(3X)**, **ReleaseMB(3X)**, **DragXYRelative(3X)**, **DragSliderValue(3X)**, **ModifierList(3X)**, **ComponentList(3X)** # $RCSfile: DragSliderValue.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:30 $

NAME

DragSliderValue — Command to drag to an value level on a Scale or ScrollBar.

SYNOPSIS

DragSliderValue *Direction Value [ModifierList]* [Button]*

DESCRIPTION

The **DragSliderValue** command will drag to a particular value on a Scale or ScrollBar. The **DragSliderValue** composed of a call to **PressMB**, a call to **LocatePointer** to move to the coordinates that represent the proper Scale or ScrollBar value, and then a call to **ReleaseMB**.

Direction Direction in which to Drag. Valid values are *Up, Down, Right, Left*.

Value Value on the slider or Scale you wish to Drag to. Value must be between **XmNminimum** and **XmNmaximum**. The literals **Min** and **Max** may be used to specify the widget's maximum and minimum respectively.

*[ModifierList]**
List of Modifiers to modify the mouse button press. For a full list of available Modifiers see **ModifierList(3X)**. The default is no modifiers.

[Button] Button to be pressed. For a full list of available Buttons see **ButtonList(3X)**. The default is Button1.

For more information about **DragSliderValue**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

DragSliderValue Up 97 <Ctrl> Btn2

This command will generate a drag from the current location (usually the scale's slider) upwards until the value of the scale is 97, using Button2 modified by the **Ctrl** key.

Note that **Min** and **Max** are reversed for a Scale and a ScrollBar. In a left-to-right environment, a Scale's **Max** is on top by default. A ScrollBar's **Max**, however, is on the bottom by default.

RELATED INFORMATION

LocatePointer(3X), **PressMB(3X)**, **ReleaseMB(3X)**, **DragComponent(3X)**, **DragXYRelative(3X)**, **ModifierList(3X)**, **ComponentList(3X)** # $RCSfile: DragXYAbs.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:31 $

NAME

DragXYAbs — Command to drag to an absolute X and Y Location.

SYNOPSIS

DragXYAbs *XCoordinate YCoordinate [ModifierList]* [Button]*

DESCRIPTION

The **DragXYAbs** command will drag to a coordinate location on the root window. The drag will begin at the cursor's current location (see **LocatePointer**). The destination coordinate is specified using the *XCoordinate* and *YCoordinate* arguments. Both coordinates are relative to the root window. The **DragXYAbs** command is composed of a call to **PressMB**, a call to **LocatePointerXYAbs** to move to the destination, and finally a call to **ReleaseMB**.

XCoordinate
: X coordinate relative to the root window.

YCoordinate
: Y coordinate relative to the root window.

*[ModifierList]**
: List of modifiers to modify the mouse button press. For a full list of available modifiers see **ModifierList(3X)**. The default is no modifiers.

[Button]
: Button to be pressed. For a full list of available buttons see **ButtonList(3X)**. The default is Button1.

For more information about **DragXYAbs**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

DragXYAbs 100 100 <Alt> Btn3

This result of this command is to Drag from the current location to (100, 100) with Button3 down and modified by Alt.

RELATED INFORMATION

LocatePointerXYAbs(3X), **PressMB(3X)**, **ReleaseMB(3X)**, **DragComponent(3X)**, **DragSliderValue(3X)**, **ModifierList(3X)**, **ComponentList(3X)** # $RCSfile: DragXYRelative.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:32 $

NAME

DragXYRelative — Command to drag in Delta X, Delta Y direction

SYNOPSIS

DragXYRelative *Direction Delta1 [Delta2]*
[ModifierList] [Button]*

DESCRIPTION

The **DragXYRelative** command will drag to a coordinate location in a widget hierarchy. The Drag will begin at the cursor's current location (see **LocatePointer**) The destination coordinate will be determined by the parameters to this function. Delta1 and Delta2 are changes in the X or Y direction (depending on the value of Direction). The **DragXYRelative** command is composed of a call to **PressMB**, a call to **LocatePointer** to move to the second coordinate, and a call to **ReleaseMB**.

Direction — Direction in which to Drag. Valid values are **Up**, **Down**, **Right**, **Left**, and **Any**. **Any** indicates a change in both the X and Y direction.

Delta1 — Change in either the X or Y direction. If *Direction* is either **Up** or **Down** then *Delta1* is used as the change in the Y direction. If *Direction* is **Right**, **Left**, or **Any**, then *Delta1* is the change in the X direction.

[Delta2] — Change in the Y direction (only used if *Direction* is **Any**).

*[ModifierList]** — List of Modifiers to modify the mouse button press. For a full list of available modifiers see **ModifierList(3X)**. The default is no modifiers.

[Button] — Button to be pressed. For a full list of available buttons see **ButtonList(3X)**. The default is Button1.

For more information about **DragXYRelative**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

DragXYRelative Any 100 -10 <Alt> Btn3

This result of this command is to Drag from the current location to (CurrentX + 100, CurrentY - 10), using Button3 modified by the **Alt** key.

RELATED INFORMATION

LocatePointer(3X), **PressMB(3X)**, **ReleaseMB(3X)**, **DragComponent(3X)**, **DragSliderValue(3X)**, **ModifierList(3X)**, **ComponentList(3X)** # $RCSfile: Exit.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:33 $

NAME

Exit — Command to exit the current test.

SYNOPSIS

Exit

DESCRIPTION

The **Exit** command will exit the current test being executed. It will do this by doing a **LocatePointer** to the cancel (**Exit**) button of the instruction box and pressing Button1. This command is usually executed when all the script information has been executed and the test is ready to finish.

For more information about **Exit**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

Exit

This will exit the current test.

RELATED INFORMATION

Continue(3X) # $RCSfile: IconMenuPost.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:34 $

NAME

IconMenuPost — Command to post the system menu on the icon window

SYNOPSIS

IconMenuPost *WidgetName*

DESCRIPTION

The **IconMenuPost** command will post the system menu on the icon window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer to the middle of the icon window, and generating a Select button press. If the icon is within the IconBox, and is not visible, scrolling in either the horizontal or vertical direction will take place within the iconBox, until the icon associated with *WidgetName* is visible.

WidgetName The XtName representing the widget to be acted upon.

For more information about **IconMenuPost**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

IconMenuPost Shell1

This command will move the pointer to the middle of the icon window associated with **Shell1**, and then generate a Select button press, thereby posting the menu.

RELATED INFORMATION

IconMove, **IconMenuSelect(3X)**, **IconMenuUnpost(3X)** # $RCSfile: IconMenuSelect.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:36 $

NAME

IconMenuSelect — Command to select an item from the icon menu

SYNOPSIS

IconMenuSelect *WidgetName* *<ItemName* | *ItemNum>*

DESCRIPTION

The **IconMenuSelect** command will select the menu item specified by *ItemName* or *ItemNum* from the icon menu of the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer down the systemMenu posted on the icon of the window to the item specified and generating a Select button click. If the icon is within the iconBox, and is not visible, scrolling in horizontal and/or vertical direction(s) is performed in the iconBox, until the icon associated with *WidgetName* is visible. If the selected item is insensitive, a warning will be printed, and the menu unposted.

WidgetName The XtName representing the widget to be acted upon.

ItemName Name of the item in the menu you wish to select.

ItemNum Number in the menu corresponding to the item you wish to select.

For more information about **IconMenuSelect**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

IconMenuSelect Shell1 Normalize

This command will move the pointer to the item **Normalize** in the icon window menu of Shell1 , and select it, thereby deiconifying the window.

IconMenuSelect Shell1 3

This command will move the pointer to item number 3 of the icon window menu of Shell1, and select it.

NOTE

In Release 1.2, the **Move** *item is removed from the icon menu. However, the* **Move** *operation is provided through the command* **IconMove**.

RELATED INFORMATION

IconMove, **IconMenuPost(3X)**, **IconMenuUnpost(3X)** # $RCSfile: IconMenuUnpost.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:37 $

NAME

IconMenuUnpost — Command to unpost the system menu on the icon window

SYNOPSIS

IconMenuUnpost *WidgetName*

DESCRIPTION

The **IconMenuUnpost** command will unpost the system menu on the icon window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer anywhere outside the icon window, and generating a Select button click.

WidgetName
: The XtName representing the widget to be acted upon.

For more information about **IconMenuUnpost**, see the **OSF/Motif Quality Assurance Test Suite User's Guide**.

EXAMPLE

IconMenuUnpost Shell1

This command will unpost the system menu on the icon window associated with Shell1.

RELATED INFORMATION

IconMove, **IconMenuPost(3X)**, **IconMenuSelect(3X)** # $RCSfile: IconMove.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:38 $

NAME

IconMove — Command to move the icon

SYNOPSIS

IconMove *WidgetName x y*

DESCRIPTION

The **IconMove** command will move the icon window associated with the specified *WidgetName* to the location specified by *x, y*. If the *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. If the icon is within the iconBox, this command does nothing. Otherwise, this function is executed by moving the pointer to the middle of the icon window, and generating a Select button press and drag until the icon window is positioned at *x, y*.

WidgetName
: The XtName representing the widget to be acted upon.

x
: X coordinate of the new position on the screen to which you wish to move the icon.

y
: Y coordinate of the new position on the screen to which you wish to move the icon.

For more information about **IconMove**, see the **OSF/Motif Quality Assurance Test Suite User's Guide**.

EXAMPLE

IconMove Shell1 100 100

This command will move the icon window associated with Shell1 to location (100, 100) on the screen, by visually dragging the icon window with the pointer. If iconBox is present, this command is a no-op.

RELATED INFORMATION

IconMenuPost(3X), **IconMenuUnpost(3X)**, **IconMenuSelect(3X)** # $RCSfile: InputString.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:39 $

NAME

InputString — Convenience routine to enter a Text string.

SYNOPSIS

InputString *"string"*

DESCRIPTION

The **InputString** command will simulate the input of a complete character string. **InputString** simply generates a call to **ClickKey** for each character in the input string. **InputString** is a shortcut for entering text into a Text or TextField widget.

string String to be input. Use **\n** for a newline, and **\t** for a Tab.

For more information about **InputString**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

InputString "Hello\nHow are\tYOU"

This will enter the string:

Hello
How are YOU

into a Text or TextField widget.

RELATED INFORMATION

LocatePointer(3X), **PressKey(3X)**, **ReleaseKey(3X)**, **KeyList(3X)**, **ModifierList(3X)**, **ClickKey(3X)** # $RCSfile: KeyList.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:40 $

NAME

KeyList — List of available keys for Automation.

DESCRIPTION

The **KeyList** is a List of the keyboard keys which can be within Automation. This list has two categories: first, a list of the Motif Virtual Keys supported by Automation, and second, a list of supported standard keyboard keys. This list complies with the standard keys supported by X and Motif. The user's .motifbind file may define additional or different virtual bindings.

Virtual Bindings:

KBackSpace, *KInsert*, *KDelete*, *KCopy*, *KCut*, *KPaste*, *KAddMode*, *KPrimaryPaste*, *KQuickPaste*, *KPageUp*, *KPageDown*, *KEndLine*, *KBeginLine*, *KActivate*, *KMenuBar*, *KClear*, *KCancel*, *KHelp*, *KMenu*, *KSelect*, *KUndo*, *KUp*, *KRight*, *KDown*, *KLeft*, *KBackTab*, *KBeginData*, *KDeselectAll*, *KEndData*, *KEnter*, *KEscape*, *KExtend*, *KNextField*, *KNextMenu*, *KPageRight*, *KPageLeft*, *KPrevField*, *KPrevMenu*, *KPrimaryCopy*, *KPrimaryCut*, *KQuickCopy*, *KQuickCut*, *KQuickExtend*, *KReselect*, *KRestore*, *KSelectAll*, *KSpace*, *KTab*

Regular Keys:

Ka - Kz, *K0 - K9*, *KSlash*, *KBackSlash*, *KBraceRight*, *KBraceLeft*, *KBracketRight*, *KBracketLeft*, *KReturn*, *KExcalmation*, *KAt*, *KNumberSign*, *KDollar*, *KPercent*, *KCaret*, *KAmpersand*, *KAsterisk*, *KParenLeft*, *KParenRight*

For more information about **KeyList**, in specific the uses of all these keys see the **VirtualBindings(3X)** reference page in the *OSF/Motif Programmer's Reference Manual*; also see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

RELATED INFORMATION

PressKey(3X), **ReleaseKey(3X)**, **ClickKey(3X)**, **ModifierList(3X)** # $RCSfile: LocatePointer.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:41 $

NAME

LocatePointer — Command to move the pointer to a specified location

SYNOPSIS

LocatePointer *WidgetName [Component] [Destination]*

DESCRIPTION

The **LocatePointer** command will move the pointer over any widget in the current widget hierarchy. If only a *WidgetName* **argument is supplied, the pointer will be moved over the midpoint of that widget. Optionally, the pointer can move to the specified component within a widget (for example, to a certain character in a text widget). The** *destination* argument to **LocatePointer** orders the automation code to move *off* of the given widget in a the specified direction. This option is very useful for synthesizing traversal within menus.

Widget The XtName of the widget to be acted upon.

[Component] Particular component within the widget. See **ComponentList(3X)** for a full list of supported components.

[Destination] *Destination* can be either an integer correponding to an index within the *component*, or one of the following literals. **Off**, **Left**, **Right**, **Down** and **Up** will move the pointer off the widget specified in the direction specified. *Off* will take the quickest path possible. In the case where the component is **RListItem**, *Destination* refers to the visible list item number (0 is the first visible element in the list; the list refers only to visible items). A value of **Max** for *Destination* in a List refers to the last visible item, while **Min** refers to the first visible item. If the component is *RTextChar*, *Destination* refers to the visible character position in the Text widget (0 is the first visible character). A value of **Max** for *Destination* in a Text widget refers to the area after the last visible character, while **Min** refers to the area before the first visible character.

For more information about **LocatePointer**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

LocatePointer List1 RListItem 2

This will move the pointer over the 3rd Item in List1.

LocatePointer Text1 RTextChar Max

This will move the pointer to the area after the last visible character in Text1.

LocatePointer *CascadeButton1 Right*

This command will move the pointer off CascadeButton1 to the right. This command is very useful in traversing menus.

LocatePointer *Pulldown1 RTearOffButton*

This command will move to the TearOff button associated with the Pulldown menu. The Pulldown menu needs to be visible and the TearOff feature must be enabled.

RELATED INFORMATION

ComponentList(3X), **LocatePointerXYAbs** # $RCSfile: LocatePointerXYAbs.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:42 $

NAME

LocatePointerXYAbs — Command to move the pointer to absolute location

SYNOPSIS

LocatePointerXYAbs *XCoordinate YCoordinate*

DESCRIPTION

The **LocatePointerXYAbs** command will move the pointer to an absolute point on the root coordinate system.

XCoordinate X Coordinate relative to the root.

YCoordinate Y Coordinate relative to the root.

For more information about **LocatePointerXYAbs**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

LocatePointerXYAbs 500 500

This will move the pointer to (500, 500) in the root window.

RELATED INFORMATION

LocatePointer(3X), DragXYAbs(3X) # $RCSfile: Manual.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:43 $

NAME

Manual — Command to enter Manual mode.

SYNOPSIS

Manual

DESCRIPTION

The **Manual** command will put the test into Manual mode. Manual mode is when events from the actual user can be sent to the current hierarchy. When Manual mode is entered, an Information Dialog will be placed at (500, 0) which is on top of the InstructionBox. To exit Manual mode, simply click on the OK button of the Information Dialog, and the test will move the pointer back to the location where it was located before Manual mode started. As soon as the OK button is clicked, the test will go back into Automated mode. Manual mode should be used as a debugging tool in conjuction with -W option. Once Automation has created you the xwd dumps of the failed image, you can run the previous version of the test (which compared properly) and use Manual to stop Automated processing at the point of failure. This way use can use xmag on the xwd dump and on the test running in Manual mode to identify the pixel failures.

EXAMPLE

Manual

This will put the test in Manual mode.

For more information about **Manual**, see the **OSF/Motif Test Suite Automation User's Guide**.

RELATED INFORMATION

$RCSfile: ModifierList.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:44 $

NAME

ModifierList — List of available Modifiers for Automation.

DESCRIPTION

The **ModifierList** is a List of the available Modifiers. Modifiers can be combined in any combination. No more than 5 different modifiers can be specified. This list is based on the modifiers supported by the *X Window System*.

<Shift> Shift Modifier.

<Lock> Lock Modifer.

<Ctrl> Control Modifer.

<Meta> Meta Modifier.

<Alt> Alt Modifier.

<Mod1> Mod1 Modifier.

<Mod2> Mod2 Modifier.

<Mod3> Mod3 Modifier.

<Mod4> Mod4 Modifier.

<Mod5> Mod5 Modifier.

For more information about **ModifierList**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

RELATED INFORMATION

PressMB(3X), **ReleaseMB(3X)**, **ClickMB**, **DragComponent(3X)**, **DragXYRelative(3X)**, **DragSliderValue(3X)**, **KeyList(3X)**, **ButtonList(3X)** # $RCSfile: PressKey.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:45 $

NAME

PressKey — Command to press a standard key.

SYNOPSIS

PressKey *[ModifierList]* Key*

DESCRIPTION

The **PressKey** command will simulate the pressing of a key on the standard keyboard (including Motif Virtual Keys). The key press maybe modified by any of the standard X modifiers (any combination). This function is usually accompanied by doing a **LocatePointer** first to a region which accepts keyboard input. Be sure that **PressKey** is executed over a widget in the current widget hierarchy.

*[ModifierList]** List of Modifiers to modify the key press. For a full list of available Modifiers see **ModifierList(3X)**. The default is no modifiers.

Button Key to be pressed. For a full list of available Keys see **KeyList(3X)**.

For more information about **PressKey**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

PressKey <Alt> KRight

This will press the KRight (right arrow) key with the modifier Alt.

RELATED INFORMATION

LocatePointer(3X), **ReleaseKey(3X)**, **ClickKey(3X)**, **KeyList(3X)**, **ModifierList(3X)** # $RCSfile: PressMB.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:46 $

NAME

PressMB— Command to press any mouse button.

SYNOPSIS

PressMB *[ModifierList]* Button*

DESCRIPTION

The **PressMB** command will simulate the pressing of a button on the standard mouse. This press maybe modified by any of the standard X modifiers (any combination). This function is usually accompanied by doing a **LocatePointer** first to a region which accepts mouse input. Be sure that **PressMB** is executed over a widget in the current widget hierarchy.

*[ModifierList]** List of Modifiers to modify the mouse button press. For a full list of available Modifiers see **ModifierList(3X)**. The default is no modifiers.

Button Button to be pressed. For a full list of available buttons see **ButtonList(3X)**.

This will press mouse Button 1, with the Control and Shift modifiers held down.

For more information about **PressMB**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

PressMB <Shift> <Ctrl> Btn1

RELATED INFORMATION

LocatePointer(3X), **ReleaseMB(3X)**, **ClickMB(3X)**, **ButtonList(3X)**, **ModifierList(3X)** # $RCSfile: ReleaseKey.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:47 $

NAME

ReleaseKey — Command to release a standard key.

SYNOPSIS

ReleaseKey *[ModifierList]* Key*

DESCRIPTION

The **ReleaseKey** command will simulate the releasing of a key on the standard keyboard (including Motif Virtual Keys). The key release maybe modified by any of the standard X modifiers (any combination). This function is usually accompanied by doing a **LocatePointer** first to a region which accepts keyboard input and then doing a **PressKey**.

*[ModifierList]** List of Modifiers to modify the key release. For a full list of available Modifiers see **ModifierList(3X)**. The default is no modifiers.

Button Key to be released. For a full list of available keys see **KeyList(3X)**.

For more information about **ReleaseKey**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

ReleaseKey <Alt> KRight

This will release the KRight (right arrow) key with the modifier Alt.

RELATED INFORMATION

LocatePointer(3X), **PressKey(3X)**, **ClickKey(3X)**, **KeyList(3X)**, **ModifierList(3X)** # $RCSfile: ReleaseMB.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:48 $

NAME

ReleaseMB — Command to release any mouse button.

SYNOPSIS

ReleaseMB *[ModifierList]* Button*

DESCRIPTION

The **ReleaseMB** command will simulate the releasing of a button on the standard mouse. This release maybe modified by any of the standard X modifiers (any combination). This function is usually accompanied by doing a **LocatePointer** first to a region which accepts mouse input and performing a **PressMB**.

*[ModifierList]** List of modifiers to modify the mouse button release. For a full list of available Modifiers see **ModifierList(3X)**. The default is no modifiers.

Button Button to be pressed. For a full list of available Buttons see **ButtonList(3X)**.

For more information about **ReleaseMB**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*

EXAMPLE

ReleaseMB <Shift> <Ctrl> Btn1

This will release mouse Button1 with Control and Shift modifiers.

RELATED INFORMATION

LocatePointer(3X), **PressMB(3X)**, **ClickMB(3X)**, **ButtonList(3X)**, **ModifierList(3X)** # $RCSfile: System.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:49 $

NAME

System — Command to call **UNIX** system calls

SYNOPSIS

System *SystemCommand [CommandArgs]**

DESCRIPTION

The **System** command will allow the user to make a **UNIX** system call within the script language. This can be used to start processes which are required to be running before a test is executed. The process may be system process such as **ps** or **kill**, or can be a user defined process such as a shell script. The **System** command is implemented by forking a child process and then executing the command via exec().

SystemCommand Name of the command you wish to execute (either system or user defined).

*[CommandArgs]** List of arguments to the command you wish to execute. Default is no arguments.

For more information about **System**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

System mwm -multiscreen

This will execute the command mwm with the argument -multiscreen.

RELATED INFORMATION

$RCSfile: Wait.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:50 $

NAME

Wait — Command to Wait an Automated script

SYNOPSIS

Wait *Seconds*

DESCRIPTION

The **Wait** command will wait in the current position for *Seconds* number of seconds. The command is very usefull when scrolling in a ScrolledList or ScrolledText widget when the widget contains a good amount of information. Since the scrolling actions are timer based, a simple **DragSliderValue** may not produce the desired effect. If you follow a **DragSliderValue** with a **Wait 10** then the scrolled widget should have enough time to fully scroll through all its data. While the program is waiting, all events will be processed.

Seconds Number of seconds to wait in the current position.

For more information about **Wait**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

Wait 10

This will cause the client to wait in its current state for 10 seconds.

RELATED INFORMATION

DragComponent(3X), **DragXYRelative(3X)**, **DragSliderValue(3X)**, **DragXYAbs(3X)** # $RCSfile: WindowDeiconify.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:51 $

NAME

WindowDeiconify — Command to deiconify a window

SYNOPSIS

WindowDeiconify *WidgetName*

DESCRIPTION

The **WindowDeiconify** command will deiconify the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. If the icon is within the IconBox, and is not visible, scrolling in horizontal and/or vertical direction(s) is performed in the iconBox, until the icon associated with *WidgetName* is visible. Actual deiconifying is executed by moving the pointer to the icon within/without the iconBox, and generating a double click of the Select button.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowDeiconify**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowDeiconify Shell1

This will move the pointer to the icon of **Shell1** and deiconify it by generating a double click of the Select button. If the icon is within the IconBox and not within view, this command will first move the pointer to horizontal and/or vertical scrollbars of the IconBox, and generate the necessary number of button clicks to scroll the icon into view.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowFocus.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:53 $

NAME

WindowFocus — Command to give input focus to a window

SYNOPSIS

WindowFocus *WidgetName*

DESCRIPTION

The **WindowFocus** command will give input focus to the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer to the middle of the title bar of the window frame, and generating a mouse Select button click.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowFocus**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowFocus Shell1

This command will move the pointer to the middle of the TitleBar of **Shell1**'s window, and generate a button click. This command is useful to set input focus on a window if the keyboardFocusPolicy is set to *explicit*.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)**, **WindowResize(3X)** # $RCSfile: WindowIconify.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:53 $

NAME

WindowIconify — Command to iconify a window

SYNOPSIS

WindowIconify *WidgetName*

DESCRIPTION

The **WindowIconify** command will iconify the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer to the minimize button of the window, and generating a mouse Select button click.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowIconify**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowIconify Shell1

This command will move the pointer to the minimize button of **Shell1** window and generate a Select button click, thereby iconifying the window.

RELATED INFORMATION

WindowRaise(3X), **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowItemCheck.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:55 $

NAME

WindowItemCheck — Command to check the existance of a window decoration or a window menu item of a window

SYNOPSIS

WindowItemCheck *WidgetName ItemGroup ItemName*

DESCRIPTION

The **WindowItemCheck** command will check whether the item specified by *ItemName* is present as part of the *ItemGroup* in the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. *ItemGroup* can be one of **RWindowDecoration**, **RWindowMenu**, or **RIconMenu**, and indicates which portion of window management will be checked. The command prints out a message saying whether the specified item is present or not.

If *ItemGroup* is specified as **RWindowDecoration**, then *ItemName* should be one of **RResizeHandles**, **RBorder**, **RTitle**, **RSystemMenu**, **RMinimize**, or **RMaximize**. If, on the other hand, *ItemGroup* is either **RWindowMenu**, or **RIconMenu**, then *ItemName* can be any character string.

WidgetName The XtName of the widget to be acted upon.

ItemGroup String defining the group indicating the part of the window management that is being checked.

ItemName String defining the actual item that is being checked for existance.

For more information about **WindowItemCheck**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowItemCheck Shell1 RWindowDecoration RMinimize

This will check the window decoration for the presence of minimize button. If the minimize button is present the message, the following message will be printed.

```
RMinimize IS present as part of decorations.
```

If RMinimize is not present, the following message will be printed.

```
RMinimize NOT present as part of decorations.
```

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowMaximize.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:56 $

NAME

WindowMaximize — Command to maximize a window

SYNOPSIS

WindowMaximize *WidgetName*

DESCRIPTION

The **WindowMaximize** command will maximize the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer to the maximize button of the window, and generating a Select button click. By default, if no maximum size is specified for the window, this will make the window fill the entire screen; otherwise, the window size is increased to its maximum size.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowMaximize**, see the **OSF/Motif Test Suite Automation User's Guide**.

EXAMPLE

WindowMaximize Shell1

This command will move the pointer to the maximize button of window associated with **Shell1** in its normal state, and generate a Select button click, thereby making the window grow to its maximum size.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowMenuPost.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:56 $

NAME

WindowMenuPost — Command to post the window menu

SYNOPSIS

WindowMenuPost *WidgetName*

DESCRIPTION

The **WindowMenuPost** command will post the window menu on the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer to the systemMenu button of the window, and generating a Select button press.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowMenuPost**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowMenuPost Shell1

This command will move the pointer to the systemMenu button on the window associated with **Shell1** and generate a Select button press, thereby posting the window menu.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowMenuSelect.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:57 $

NAME

WindowMenuSelect — Command to select an item from the window menu

SYNOPSIS

WindowMenuSelect *widgetname* *<ItemName | ItemNum>*

DESCRIPTION

The **WindowMenuSelect** command will select the menu item specified by *ItemName* or *ItemNum* from the window menu on the window associated with *widgetname*. If *widgetname* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer down the systemMenu to the item specified and generating a Select button click. If the selected item is insensitive, a warning will be printed and the menu unposted.

widgetname The XtName of the widget to be acted upon.

ItemName Name of the item in the menu you wish to select.

ItemNum Number in the menu corresponding to the item you wish to select.

For more information about **WindowMenuSelect**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowMenuSelect Shell1 Lower

This command will move the pointer to the item **Lower** in the window menu of the window associated with **Shell1**, and select it.

WindowMenuSelect Shell1 3

This command will move the pointer to item number 3 of the window menu of the window associated with **Shell1**, and select it.

NOTE

In version 1.2, items **Move** *and* **Size** *are removed from the window menu. However, the corresponding operations are provided through the commands* **WindowMove** *and* **WindowResize** *respectively*.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeIconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnPost(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowMenuUnpost.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:58 $

NAME

WindowMenuUnpost — Command to unpost the window menu

SYNOPSIS

WindowMenuUnpost *WidgetName*

DESCRIPTION

The **WindowMenuUnpost** command will unpost the window menu on the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer anywhere outside the system menu button of the window, and generating a Select button click.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowMenuPost**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowMenuUnpost Shell1

This command will unpost the window menu of the window associated with **Shell1**.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus** # $RCSfile: WindowMove.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:41:59 $

NAME

WindowMove — Command to move a window

SYNOPSIS

WindowMove *WidgetName x y*

DESCRIPTION

The **WindowMove** command will move the window associated with *WidgetName* to the location *x, y*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer to the middle of the titlebar of the window, generating a mouse Select button press and drag until the window is positioned at *x, y*. When using **WindowMove** with a TearOff Menu, *WidgetName* should be one of the widgets contained in the original Pulldown or PopupMenu.

WidgetName The XtName of the widget to be acted upon.

x X coordinate of the new position on the screen to which you wish to move the window.

y Y coordinate of the new position on the screen to which you wish to move the window.

For more information about **WindowMove**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowMove Shell1 100 100

This command will move the window associated with Shell1 to location (100, 100) on the screen, by visually dragging the window with the pointer.

WindowMove pushbutton 100 100

If the Pulldown or PoupMenu that contains pushbutton has just been torn off, this command will move the TearOff window to location (100, 100) on the screen, by visually dragging the window with the pointer.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowNormalize.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:42:01 $

NAME

WindowNormalize — Command to bring a window to its normal size

SYNOPSIS

WindowNormalize *WidgetName*

DESCRIPTION

The **WindowNormalize** command will bring the window associated with *WidgetName*, to its normal size from its maximized state. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This function is executed by moving the pointer to the maximize button of the window in maximized state, and generating a mouse Select button click.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowNormalize**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowNormalize Shell1

This command will move the pointer to the maximize button of the window associated with **Shell1** in its maximized state, and generate a Select button click, thereby making the window go back to its normal size.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowMove(3X) WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowRaise.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:42:02 $

NAME

WindowRaise — Command to raise a window to the top of window stack

SYNOPSIS

WindowRaise *WidgetName*

DESCRIPTION

The **WindowRaise** command will raise the window associated with *WidgetName* to the top of the window display stack. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used.

WidgetName The XtName of the widget to be acted upon.

For more information about **WindowRaise**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowRaise Shell1

This command will raise the window associated with **Shell1** to the top of the display stack.

RELATED INFORMATION

WindowIconify(3X), **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)** **WindowResize(3X)**, **WindowFocus(3X)** # $RCSfile: WindowResize.3X,v $ $Revision: 1.1.2.1 $ $Date: 1994/02/03 22:42:03 $

NAME

WindowResize — Command to resize the window

SYNOPSIS

WindowResize *WidgetName gravity dimension [dimension]*

DESCRIPTION

The **WindowResize** command will resize the window associated with *WidgetName*. If *WidgetName* is not a shell with its own window on the display, the window of the nearest shell parent in the widget hierarchy will be used. This command is executed by moving the pointer to one of the resize handles of the window, pressing mouse Select button, and dragging it until the window has the desired size.

The argument *gravity* specifies which resize handle will be used to resize the window. **North, South, East, West, NorthEast, NorthWest, SouthEast, SouthWest** are legal values for *gravity*. If *gravity* is **North** or **South**, the first *dimension* is taken as the new height for the window. If *gravity* is **East** or **West**, the first *dimension* is taken as the new width of the window. If *gravity* is **NorthEast**, **NorthWest**, **SouthEast** or **SouthWest**, the first *dimension* is taken as the new width of the window, and the (optional) second *dimension* is taken as the new height of the window.

WidgetName The XtName of the widget to be acted upon.

gravity String defining the gravity which is used to determine which resize handle will be used and how the *dimension* argument(s) will be interpreted.

dimension Number defining the new size in any one direction. If *gravity* is specified as either **North** or **South**, *dimension* will be interpreted as the new height of the window; otherwise, it will be interpreted as the new width of the window.

[dimension] Number defining the new height of the window if *gravity* is **NorthEast**, **NorthWest**, **SouthEast** or **SouthWest**; otherwise this is ignored.

For more information about **WindowResize**, see the *OSF/Motif Quality Assurance Test Suite User's Guide*.

EXAMPLE

WindowResize Shell1 SouthEast 200 250

This will move the pointer to the resize handle at the lower right corner of the window associated with Shell1 and generate diagonal drag with Select button down until the window's size is 200x250.

WindowResize Shell1 South 300

This will move the pointer to the middle of the bottom resize handle of the window associated with Shell1 and generate vertical drag with Select button down, until the height of the window is 300.

WindowResize Shell1 West 300

This will move the pointer to the middle of the left resize handle of the window associated with Shell1 and generate horizontal drag with Select button down, until the width of the window is 300.

WindowResize Shell1 SouthWest 400

This will move the pointer to the resize handle at the bottom left corner of the window associated with Shell1 and generate horizontal drag with Select button down, until the width of the window is 400.

WindowResize Shell1 North 450 500

This will move the pointer to the middle of the top resize handle of the window associated with Shell1 and generate vertical drag with Select button down, until the height of the window is 450.

The second *dimension*, 500, is ignored for this gravity.

RELATED INFORMATION

WindowRaise(3X), **WindowIconify(3X)**, **WindowDeiconify(3X)**, **WindowMenuPost(3X)**, **WindowMenuUnpost(3X)**, **WindowMenuSelect(3X)**, **WindowMaximize(3X)**, **WindowNormalize(3X)**, **WindowMove(3X)**, **WindowFocus(3X)**