NAME

uwm - Window Manager Client Application of X

SYNTAX

uwm [-f *filename*]

DESCRIPTION

The *uwm* command is a window manager client application of the window server.

When the command is invoked, it traces a predefined search path to locate any *uwm* startup files. If no startup files exist, *uwm* initializes its built-in default file.

If startup files exist in any of the following locations, it adds the variables to the default variables. In the case of contention, the variables in the last file found override previous specifications. Files in the *uwm* search path are:

/usr/new/lib/X/uwm/system.uwmrc
$HOME/.uwmrc

To use only the settings defined in a single startup file, include the variables, **resetbindings**, **resetmenus**, **resetvariables** at the top of that specific startup file.

ARGUMENTS

-f *filename*
Names an alternate file as a *uwm* startup file.

STARTUP FILE VARIABLES

Variables are typically entered first, at the top of the startup file. By convention, **resetbindings**, **resetmenus**, and **resetvariables** head the list.

autoselect/**noautoselect**
places menu cursor in first menu item. If unspecified, menu cursor is placed in the menu header when the menu is displayed.

delta=*pixels* indicates the number of pixels the cursor is moved before the action is interpreted by the window manager as a command. (Also refer to the **delta** mouse action.)

freeze/**nofreeze** locks all other client applications out of the server during certain window manager tasks, such as move and resize.

grid/**nogrid** displays a finely-ruled grid to help you position an icon or window during resize or move operations.

hiconpad=*n* indicates the number of pixels to pad an icon horizontally. The default is five pixels.

hmenupad=*n* indicates the amount of space in pixels, that each menu item is padded above and below the text.

iconfont=*fontname*
names the font that is displayed within icons. Font names are listed in the font directory, */usr/new/lib/X/font*.

maxcolors=*n* limits the number of colors the window manager can use in a given invocation. If set to zero, or not specified, *uwm* assumes no limit to the number of colors it can take from the color map. **maxcolors** counts colors as they are included in the file.

normali/**nonormali**
places icons created with **f.newiconify** within the root window, even if it is placed partially off the screen. With **nonormali** the icon is placed exactly where the cursor leaves it.

normalw/**nonormalw**
places window created with **f.newiconify** within the root window, even if it is placed partially off the screen. With **nonormalw** the window is placed exactly where the cursor leaves it.

push=*n* moves a window *n* number of pixels or a relative amount of space, depending on whether **pushabsolute** or **pushrelative** is specified. Use this variable in conjunction with **f.pushup, f.pushdown, f.pushright**, or **f.pushleft**.

pushabsolute/**pushrelative**
pushabsolute indicates that the number entered with push is equivalent to pixels. When an f.push (left, right, up, or down) function is called, the window is moved exactly that number of pixels.

pushrelative indicates that the number entered with the push variable represents a relative number. When an f.push function is called, the window is invisibly divided into the number of parts you entered with the push variable, and the window is moved one part.

resetbindings, **resetmenus**, and **resetvariables**
resets all previous function bindings, menus, and variables entries, specified in any startup file in the *uwm* search path, including those in the default environment. By convention, these variables are entered first in the startup file.

resizefont=*fontname*
identifies the font of the indicator that displays in the corner of the window as you resize windows. See the */usr/new/lib/X/font* directory for a list of fonts.

reverse/**noreverse**
defines the display as black characters on a white background for the window manager windows and icons.

viconpad=*n* indicates the number of pixels to pad an icon vertically. Default is five pixels.

vmenupad=*n* indicates the amount of space in pixels that the menu is padded on the right and left of the text.

volume=*n* increases or decreases the base level volume set by the *xset(1)* command. Enter an integer from 0 to 7, 7 being the loudest.

zap/**nozap** causes ghost lines to follow the window or icon from its previous default location to its new location during a move or resize operation.

BINDING SYNTAX

"*function*=[*control key(s)*]:[*context*]:*mouse events*:*" menu name "*

Function and mouse events are required input. Menu name is required with the *f.menu* function definition only.

Function

f.beep emits a beep from the keyboard. Loudness is determined by the volume variable.

f.circledown causes the top window that is obscuring another window to drop to the bottom of the stack of windows.

f.circleup exposes the lowest window that is obscured by other windows.

f.continue releases the window server display action after you stop action with the **f.pause** function.

f.focus directs all keyboard input to the selected window. To reset the focus to all windows, invoke *f.focus* from the root window.

f.iconify when implemented from a window, this function converts the window to its respective icon. When implemented from an icon, f.iconify converts the icon to its respective window.

f.lower lowers a window that is obstructing a window below it.

f.menu invokes a menu. Enclose ‘menu name’ in quotes if it contains blank characters or

parentheses.

f.menu=[*control key(s)***]:[***context* **]:***mouse events***:"** *menu name* **"**

f.move moves a window or icon to a new location, which becomes the default location.

f.moveopaque moves a window or icon to a new screen location. When using this function, the entire window or icon is moved to the new screen location. The grid effect is not used with this function.

f.newiconify allows you to create a window or icon and then position the window or icon in a new default location on the screen.

f.pause temporarily stops all display action. To release the screen and immediately update all windows, use the **f.continue** function.

f.pushdown moves a window down. The distance of the push is determined by the push variables.

f.pushleft moves a window to the left. The distance of the push is determined by the push variables.

f.pushright moves a window to the right. The distance of the push is determined by the push variables.

f.pushup moves a window up. The distance of the push is determined by the push variables.

f.raise raises a window that is being obstructed by a window above it.

f.refresh results in exposure events being sent to the window server clients for all unobscured or partially obscured windows. The windows will not refresh correctly if the exposure events are not handled properly.

f.resize resizes an existing window. Note that some clients, notably editors, react unpredictably if you resize the window while the client is running.

f.restartn causes the window manager application to restart, retracing the *uwm* search path and initializing the variables it finds.

Control Keys

By default, the window manager uses meta as its control key. It can also use ctrl, shift, lock, or null (no control key). Control keys must be entered in lower case, and can be abbreviated as: c, l, m, s for ctrl, lock, meta, and shift, respectively.

You can bind one, two, or no control keys to a function. Use the bar (|) character to combine control keys.

Note that client applications other than the window manager use the shift as a control key. If you bind the shift key to a window manager function, you can not use other client applications that require this key.

Context

The context refers to the screen location of the cursor when a command is initiated. When you include a context entry in a binding, the cursor must be in that context or the function will not be activated. The window manager recognizes the following four contexts: icon, window, root, (null).

The root context refers to the root, or background window, A (null) context is indicated when the context field is left blank, and allows a function to be invoked from any screen location. Combine contexts using the bar (|) character.

Mouse Buttons

Any of the following mouse buttons are accepted in lower case and can be abbreviated as l, m, or r, respectively: left, middle, right.

With the specific button, you must identify the action of that button. Mouse actions can be:

down function occurs when the specified button is pressed down.

up function occurs when the specified button is released.

delta indicates that the mouse must be moved the number of pixels specified with the delta variable before the specified function is invoked. The mouse can be moved in any direction to satisfy the delta requirement.

MENU DEFINITION

After binding a set of function keys and a menu name to **f.menu**, you must define the menu to be invoked, using the following syntax:

menu = " *menu name* **" {**
"*item name*" : "*action*"
.
.
.
}

Enter the menu name exactly the way it is entered with the **f.menu** function or the window manager will not recognize the link. If the menu name contains blank strings, tabs or parentheses, it must be quoted here and in the f.menu function entry. You can enter as many menu items as your screen is long. You cannot scroll within menus.

Any menu entry that contains quotes, special characters, parentheses, tabs, or strings of blanks must be enclosed in double quotes. Follow the item name by a colon (:).

Menu Action

Window manager functions

Any function previously described. E.g., **f.move** or **f.iconify**.

Shell commands

Begin with an exclamation point (!) and set to run in background. You cannot include a new line character within a shell command.

Text strings

Text strings are placed in the window server's cut buffer.

Strings with a new line character must begin with an up arrow (^), which is stripped during the copy operation.

Strings without a new line must begin with the bar character (|), which is stripped during the copy operation.

Color Menus

Use the following syntax to add color to menus:

menu = "*menu name***" (***color1:color2:color3:color4***) {**
"*item name*" : (*color5 :color6*) : " *action* "
.
.
.
}

color1 Foreground color of the header.

color2 Background color of the header.

color3 Foreground color of the highlighter, the horizontal band of color that moves with the cursor within the menu.

color4 Background color of the highlighter.

color5 Foreground color for the individual menu item.

color6 Background color for the individual menu item.

Color Defaults

Colors default to the colors of the root window under any of the following conditions:

1) If you run out of color map entries, either before or during an invocation of *uwm*.

2) If you specify a foreground or background color that does not exist in the RGB color database (*/usr/lib/rgb.txt*) both the foreground and background colors default to the root window colors.

3) If you omit a foreground or background color, both the foreground and background colors default to the root window colors.

4) If the total number of colors specified in the startup file exceeds the number specified in the *maxcolors* variable.

5) If you specify no colors in the startup file.

EXAMPLES

The following sample startup file shows the default window manager options:

```
# Global variables
#
resetbindings;resetvariables;resetmenus
autoselect
delta=25
freeze
grid
hiconpad=5
hmenupad=6
iconfont=oldeng
menufont=timrom12b
resizefont=9x15
viconpad=5
vmenupad=3
volume=7
#
# Mouse button/key maps
#
# FUNCTION    KEYS  CONTEXT  BUTTON    MENU(if any)
# ========    ====  =======  ======    ============
f.menu =      meta  :    :left down   :"WINDOW OPS"
f.menu =      meta  :    :middle down :"EXTENDED WINDOW OPS"
f.move =      meta  :w|i  :right down
f.circleup =  meta  :root :right down
#
# Menu specifications
#
menu = "WINDOW OPS" {
"(De)Iconify":     f.iconify
Move:              f.move
Resize:            f.resize
Lower:             f.lower
Raise:             f.raise
}

menu = "EXTENDED WINDOW OPS" {
Create Window:                    !"xterm &"
Iconify at New Position:    f.lowericonify
Focus Keyboard on Window:         f.focus
Freeze All Windows:               f.pause
Unfreeze All Windows:             f.continue
Circulate Windows Up:             f.circleup
Circulate Windows Down:           f.circledown
}
```

RESTRICTIONS

The color specifications have no effect on a monochrome system.

FILES

/usr/lib/rgb.txt
/usr/new/lib/X/font
/usr/skel/.uwmrc
/usr/new/lib/X/uwm/system.uwmrc
$HOME/.uwmrc

SEE ALSO

X(1), X(8C)

AUTHOR

M. Gancarz, DEC Ultrix Engineering Group, Merrimack, New Hampshire, using some algorithms originally by Bob Scheifler, MIT Laboratory for Computer Science