WindowMaker

X Window Manager

User's Guide

Alfredo K. Kojima

Version 0.10.0

Contents

This manual describes the usage and configuration of the WindowMaker window manager. It is intended for both users who never used the X Window System and for users who have experience with other window managers.

How to Read this guide

If you never have used a X window manager, you should read all this guide, as it contains detailed instructions for newbies.

Text in sans serif font, indicate instructions you must follow to accomplish a given task. If you're out of time (or patience), you should at least read text in these parts.

You can ignore the text in Extra Bindings boxes while you're getting familiar with WindowMaker. Once you've got familiar with it, you can read the text in these boxes to learn more ways to accomplish tasks.

Chapter 1

Introduction

1.1 What is a window manager?

If you come from the Windows®or MacOS™world, you might be confused about all these things like window managers, X windows etc.

In the UNIX™world, the task of providing a graphical user interface (GUI) is normally divided by 3 different components:

- the window server;
- the window manager and
- the user interface toolkit.

The **window server** is standard and is usually the *X Window System* or some vendor provided compatible version of it. The X Window System, or X for short, is a window server. It's function is to provide a portable and high-level access to devices like keyboard, mouse and video display. It allows applications to show graphical information on the display through rectangular areas called windows.

Most user interface objects, like buttons, menus and scrollers are made of windows. The top level windows displayed by applications are named windows as well. These objects are not provided by the window server. These must be made by the application program or by the user interface toolkit.

For more information, read the manual page for `X(1)` and the documentation for Xlib.

The primary function of the **window manager** is to control the layout of top level windows on screen. WindowMaker is a window manager. It provides a titlebar and a resizebar to change window layout, application menus to launch applications and execute special commands, application icons, miniwindows and an application dock. They will be explained in more detail in the following chapters.

The **user interface toolkit** is a library or collection of libraries that provide an API for application developers to program the interfaces for their applications. Toolkits generally provide controls like buttons, menus, radio-buttons etc to be used for program interaction. There is currently many of these toolkits available for X. Motif™, OpenLook™, and Athena are examples of toolkits.

All other features normally found in other operating systems, like file managers, are implemented as separate programs and are not directly related to the window manager.

Chapter 2

The Workspace

2.1 Working with Menus

Menus provide a list of commands that you can execute. To execute a command listed in a menu, click in the corresponding item. The item will blink telling that the command is going to be executed.

Grayed commands are disabled and cannot be executed at that moment. If you click on them nothing will happen.

Some menu entries have a little triangular indicator at the right. Selecting these entries will open a submenu, with a new list of commands.

You can use the keyboard to traverse and execute commands in some of the menus. First you must hit the key used to open the menu — like `F12` for the root menu — to enable keyboard traversal of it. Then you can use the Up and Down arrow keys to change the current selected item and the Left and Right arrow keys to jump between submenus and parent menus. To execute the current selected item press `Return`. To close the menu or stop menu traversal, press `Escape`. Additionally, pressing the first letter for an menu item, will jump the current selection to that item.

You can make frequently used menus "stick" to the workspace by dragging the titlebar of the menu. This will make a close button appear in the menu titlebar. If you want to close the menu, just click in that button.

Menus are normally placed on top of other windows and cannot be obscured by them. If you want the menus to be able to be obscured by lowering them, double click the menu titlebar while holding the `Meta` key. Repeat this to make the menus not obscurable again.

2.1.1 The Root Window Menu

The *Root Window Menu* or *Applications Menu* has items that allow you to quickly launch applications and do some workspace management.

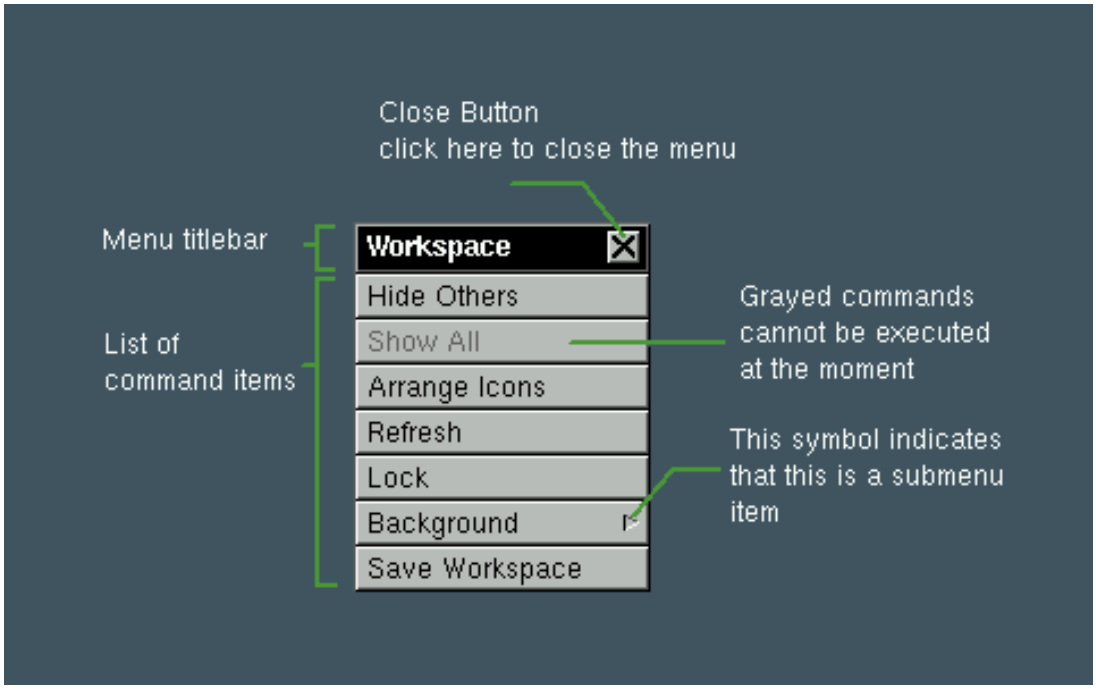
To open this menu, click on the workspace (root window) with the 3rd mouse button or hit the key bound to it (`F12` by default).

The contents of the applications menu can be configured to hold the applications installed on your system. To learn how to configure it, read the section on application menu configuration.

2.1.2 The Workspaces Menu

The *Workspaces Menu* allows you to create, switch, destroy and rename workspaces.

It has the following items:



New creates a new workspace and automatically switches to it

Destroy Last destroys the last workspace unless it is occupied by some window

Workspaces Each workspace has a corresponding item in the Workspaces menu. Clicking in one of these entries will switch from the current workspace to the selected workspace.

The current active workspace is indicated by a small indicator at the left of the workspace item.

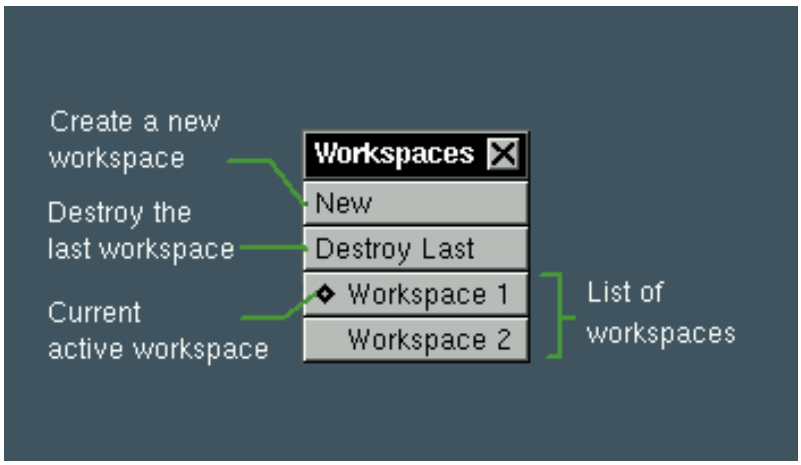


Figure 2.1: Workspaces Menu

To change the name of a workspace you must first "stick" the menu. Then `Control` click in the item corresponding to the workspace you want to rename. The item will turn into a

editable text field where you can edit the workspace name. To finish editing the workspace name, press `Return`; to cancel itś, press `Escape`.

There is a limit of 16 characters on the length of the workspace name.

Figure 2.2: The editable workspace menu

2.1.3 The Window Commands Menu

2.1.4 The Window List Menu

2.2 Working with Applications

2.2.1 Starting an Application

2.2.2 Hiding an Application

2.2.3 Docking an Application

2.3 Working with Workspaces

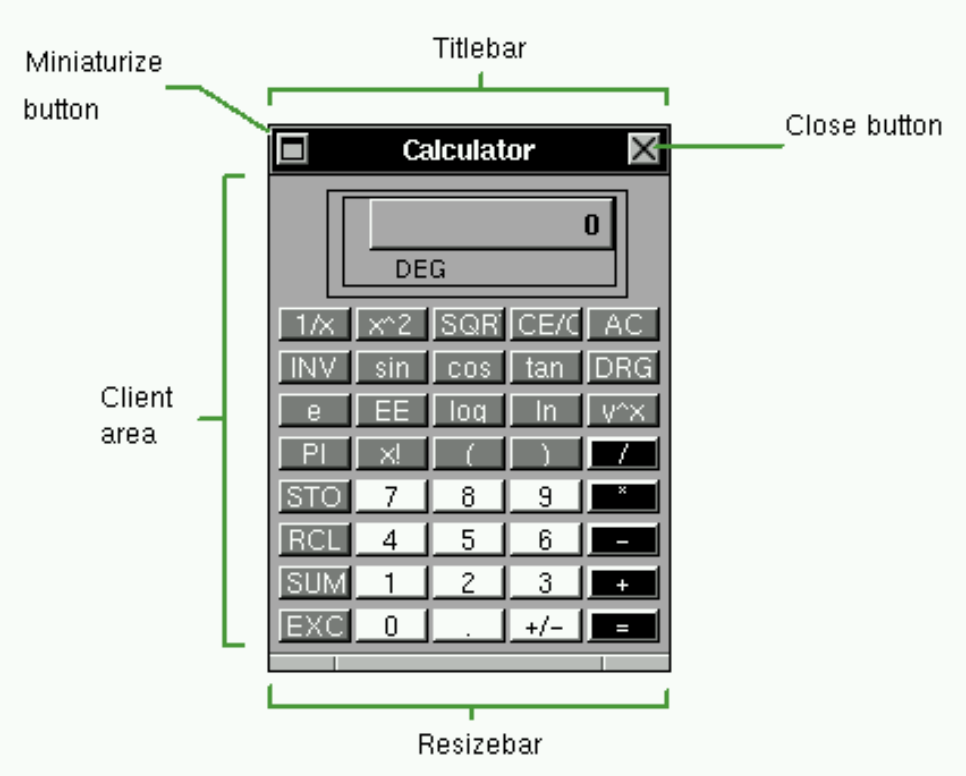
2.3.1 The Pager

Chapter 3

Windows

3.1 Anatomy of a Window

Generally, an application window will have the following layout:



Titlebar. The titlebar presents the name of the application, document or window. It's color indicates the keyboard focus state and type of the window. You can use it to move, activate, raise, lower and access the window commands menu.

Miniaturize button. You can click on the miniaturize button to miniaturize/iconify a window or click it with the `Meta` key pressed to hide the application.

Close button. The close button can be used to close a window or kill the application, if the application can't understand the close message.

Resizebar. You use the resizebar to (surprise!) resize a window.

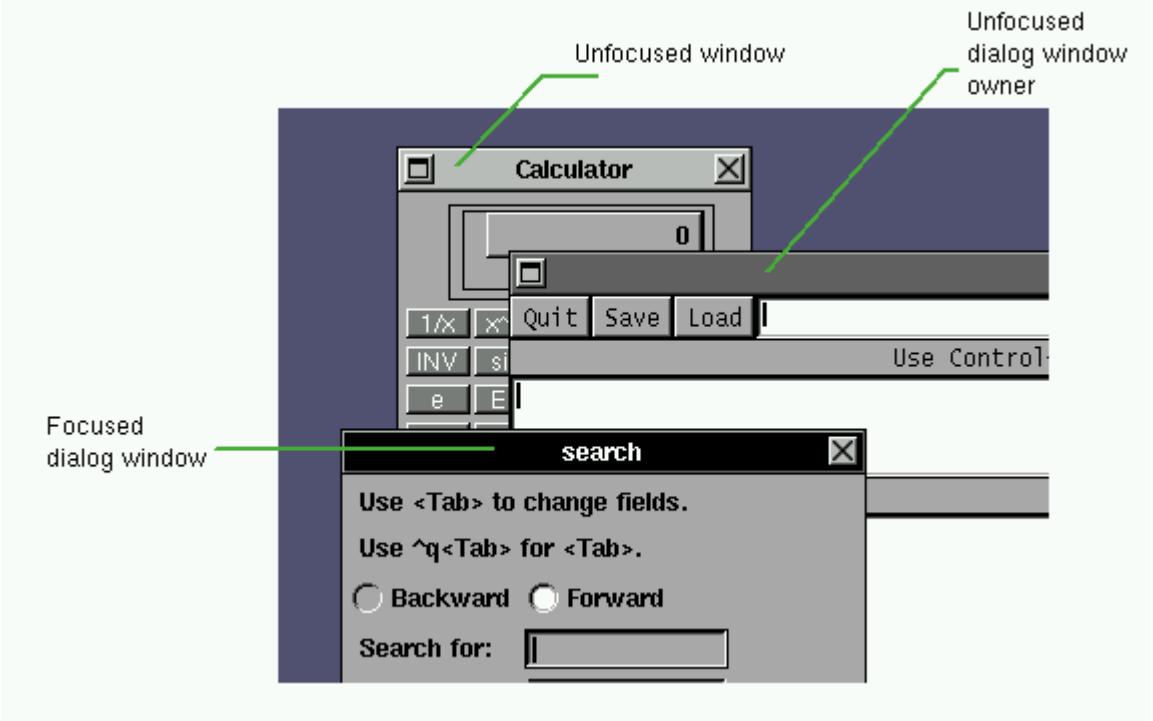
Client Area. The client area is where the application show it's information. If the window if inactive, you can click on it to activate it.

3.2 Working with Windows

3.2.1 Focusing a Window

Windows can be in two states: *focused*, or *unfocused*. The focused window (also called the key or active window) has a black titlebar and is the window that receives keyboard input, ie: where you can type text. Usually it's the window where you work on. Only one window may be focused at a time. Unfocused windows have a light gray titlebar.

Some applications have a special type of window, called dialog windows transient windows or panels. When these windows are active, the window that owns them (the main window) get a dark gray titlebar. As soon as the dialog window is closed, the focus is returned to the owner window. **??** shows an active Open File panel and it's owner window.



There are three styles of window focusing:

Click-to-Focus, or manual focus mode. In click-to-focus mode, you explicitly choose the window that should be focused. This is the default mode.

Focus-Follow-Mouse, or auto-focus mode. In this mode, the focused window is chosen based on the position of the mouse pointer. The window below the mouse pointer is always the focused window.

Sloppy-Focus, or semiauto-focus mode. This is similar to the focus-follow-mouse mode, but if you move the pointer from a window to the root window, the window will not loose focus.

You can choose between these modes with the focusmode_op option.

To focus a window in click-to-focus mode:

- Click on the titlebar, resizebar or in the client area of the window with the left or right mouse button.
  OR
- Click on the titlebar with the middle mouse button. This will focus the window without bringing it to the front.
  OR
- Open the window list menu and select the window to focus.

When you click in the client area of an inactive window to set the focus, the click is normally processed by the application. If you find this behaviour a little confusing, you can make the application ignore this click by using the ignorefocusclick_op option.

To focus a window in focus-follow-mouse mode:

- Move the pointer over the window you want to focus.

3.2.2 Reordering Overlapping Windows

Windows can overlap other windows, making some windows be over or in front of others.

To bring a window to the front:

- Click on the titlebar or resizebar of the desired window with the left mouse button.
  OR
- Select the desired window from the Window List menu.

Dialog/transient windows are always placed over their owner windows, unless the ontop-transients_op option is disabled. Some windows have a special attribute that allow them be permanently over normal windows. You can make specific windows have this attribute use the alwaysontop_op window option or set it in the Window Inspector panel.

Extra Bindings

Action	Effect
Meta-Click on the window titlebar with the left mouse button	Sends the window to the back.
Meta-Click on the Client Area of the window with the left mouse button	Brings the window to the front and focuses it.
Hold the `Meta` key and press the Up Arrow key	Brings the current focused window to the front.
Hold the `Meta` key and press the Down Arrow key	Sends the current focused window to the back.

3.2.3 Moving a Window

To move the window around the screen, drag the window through it's titlebar with the left mouse button pressed. This will also bring the window to the front and focus the window.

To move a window:

- Click on the titlebar of the window you want to move with the left mouse button and drag it with the button pressed.

While you move the window, a little box will appear in the screen, indicating the current window position in pixels, relative to the top left corner of the screen. You can change the location of this position box by hitting the `Shift` key during the move operation.

In some rare occasions, it is possible for a window to be placed off screen. This can happen with some buggy applications. To bring a window back to the visible screen area, select the window in the Window List menu. You can prevent windows from doing that with the dontmoveoff_at window attribute.

Extra Bindings

Action	Effect
Drag the titlebar with the middle mouse button	Move the window without changing it's stacking order.
Drag the titlebar while holding the `Control` key	Move the window without focusing it.
Drag the client area or resizebar while holding the `Meta` key	Move the window.

3.2.4 Resizing a Window

The size of a window can be adjusted by dragging the resizebar.

Depending on the place you click to drag the resizebar, the resize operation is constrained to a direction.

To resize a window:

- To change the window's height, click in the middle region of the resizebar and drag it vertically.
- To change the window's width, click in either end regions of the resizebar and drag it horizontally.
- To change both height and width at the same time, click in either end regions of the resizebar and drag it diagonally.

While you resize the window, a little box will appear in the screen, indicating the current window size. You can change the location of this size box or change it into a different format by hitting the `Shift` key during the resize operation.

If a window gets too big to fit on the screen and you are loose access to it's titlebar or resizebar, you can move the window through the client are, holding the `Meta` key and then resize it.

Extra Bindings

Action	Effect
Drag the window in the client area with the Right mouse button, while holding the `Meta` key	Resizes the window.
Drag the resizebar with the middle mouse button	Resize the window without bringing it to the front
Drag the resizebar while holding the `Control` key	Resize the window without focusing it.

3.2.5 Miniaturizing a Window

If you want to temporarily get rid of a window, you can miniaturize it. When miniaturizing a window, it will shrink into a miniwindow with a icon and a title that is placed at the bottom of the screen.

Figure 3.1: Miniwindow

You can move the miniwindow around the screen by dragging it. Unlike application icons, miniwindows cannot be docked.

To restore a window from it's miniwindow, double click the miniwindow. The window will be restored in the current workspace, with the same position, size and contents as it had before miniaturization.

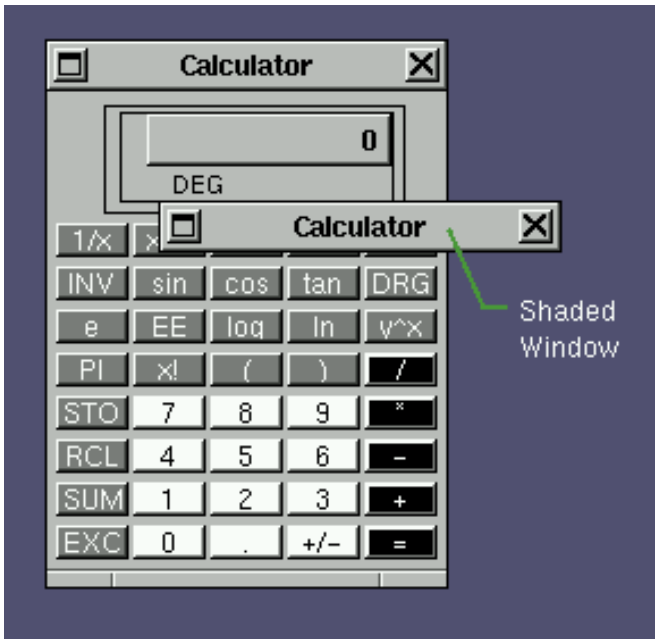
To miniaturize a window:

- Click on the miniaturize button.

To restore a miniaturized window:

- Double click in the miniwindow.

You can also restore all miniaturized and hidden windows of a given application by double clicking in it's application icon with the middle mouse button.



3.2.6 Shading a Window

If you want to temporarily get rid of a window, an option for it's miniaturization is to *shade* it. When you shade a window, the window rolls up to it's titlebar. You can do almost everything you do with a normal window with shaded windows, like miniaturizing or closing it.

To shade a window:

- Double Click on the titlebar of the window.

3.2.7 Closing a Window

After finishing work in a window, you can close it to completely get rid of it. When you close a window, it is removed from the screen and can no longer be restored. So, before closing a window, be sure you have saved any work you were doing on it.

Some windows will have a close button with some dots around it. These windows can't be closed normally and the only way to get rid of them is by exiting the application. You should try exiting from inside the application (through it's menus or buttons) when possible. Otherwise you can force WindowMaker to "kill" the application.

To force the closure of a window (by killing the application):

- Hold the `Control` key and click on the close button.
  OR
- Double click the close button.

It is also possible to kill applications that can be normally closed by clicking the close button while holding the `Control` key.

3.2.8 Maximizing a Window

If you want to resize a window to occupy the whole screen, you can maximize the window. When you unmaximize it, the window will be restored to the same position and size it was before maximized.

To maximize a window:

- Hold the `Control` key and double click on the window titlebar to resize the window's height to full screen.
- Hold the `Shift` key and double click on the window titlebar to resize the window's width to full screen.
- Hold both the `Control` and `Shift` keys and double click on the window titlebar to resize both window's height and width to full screen.

To restore the size of a maximized window:

- Hold the `Control` OR `Shift` key and double click on the window titlebar.

You can select whether the window should be maximized to the whole screen or if the position of the Dock should be accounted for by setting the windock_op option.

Chapter 4

Configuring WindowMaker

4.1 The Defaults System

4.1.1 Property List File Format

Configurations and other data that must be kept between sessions (like the applications that were docked) are stored in the format of *property lists*in the ~/GNUstep/Defaults directory. The syntax of a property list is simple, but if you need to change it manually, you must take care to not leave any syntax errors.

Bacchus Naur Form (BNF) for the property list

```
<object>        ::=  <string> | <data> |<array> | <dictionary>
<string>        ::=  text with non-alphanumeric characters | alphanumeric text
<array>         ::=  '(' [ <object> { ',' <object> }* ] ')'
<dictionary>    ::=  '{' [ <keyval_pair> { ','<keyval_pair> }*] '}'
<keyval_pair>   ::=  <string> '=' <object> ';'
```

Example property list file:

```
{
        "*" = {
                Icon = "defaultAppIcon.xpm";
        };
        "xterm.XTerm" = {
                Icon = "xterm.xpm";
        };
        xconsole = {
                Omnipresent = YES;
                NoTitlebar = YES;
                KeepOnTop = NO;
        };
}
```

The property list above is a dictionary with 3 dictionaries inside. The first is keyed by "*", the second by "XTerm.xterm" and the last by "xconsole".

Note that all strings that have non-alphabetic or numeric characters (like a dot "." or the asterisk "*" are enclosed by double quotes. Strings with only alphanumeric characters may or may not be enclosed in double quotes, as they will not make any difference.

Here is another example:

```
{
        FTitleBack = ( hgradient, gray, "#112233" );
}
```

The property list in the example above contains an array with 3 elements with a key named "FTitleBack".

Except for cases like file names and paths, all value strings are case insensitive, i.e.: YES = Yes = yes = yEs

4.1.2 Preferences

General preference options are stored in the `~/GNUstep/Defaults/WindowMaker` file.

Note that values marked as *Default* are values that are assumed if the option is not specified, instead of *factory default* values that are set in the preference file.

General Configuration

Option	Values	Default Value	Description
PixmapPath	list of directories separated by :	depends on the system	A list of directories where pixmaps can be found. The pixmaps for things like icons, are searched in these paths in order of appearance.
NoDithering	YES or NO	NO	Disable internal dithering of images. Not recommended for displays with less than 8 bits per pixel.
ColormapSize	integer number > 1	4	Number of colors for each of the red, green and blue components to be used for the dithering colormap. This value must be greater than 1 and smaller than 6 for 8bpp displays. It only makes sense on PseudoColor displays. This option has not effect on TrueColor displays. Larger values result in better appearance, but leaves less colors for other applications
ModifierKey	modifier key name	Mod1	The key to use as the modifier being referred as `Meta` in this manual, like `Meta` dragging a window to move it. Valid values are `Alt`, `Meta`, `Super`, `Hyper`, `Mod1`, `Mod2`, `Mod3`, `Mod4`, `Mod5`.
UseSaveUnders	YES or NO	NO	Use *saveunders* in WindowMaker windows. This can improve performance but increases memory usage. It also can cause problems with refreshing in some applications.

Preference Options

Option	Values	Default Value	Description
IconSize	integer number > 4	64	The size of application icons and miniwindows.
FocusMode	Manual or ClickToFocus, Auto or FocusFollowsMouse, SemiAuto or Sloppy	Manual	The mode of input focus setting. Refer to the section about focus selection (??) for an explanation of the focus modes.
AutoFocus	YES or NO	YES	Whether windows should receive input focus automatically when they are first shown on screen. It can be annoying if a window is shown while you are typing something in another window.
IgnoreFocusClick	YES or NO	NO	Whether the mouse click used to focus a window should be ignored or treated normally.
RaiseDelay	integer number	0	How many tenths of a second to wait before raising a window in Auto or SemiAuto focus mode. 0 disables this feature.
ColormapMode	Manual or ClickToFocus, Auto or FocusFollowsMouse	Auto	The mode of colormap setting. In *Manual* or *ClickToFocus* mode, the colormap is set to the one belonging to the current focused window. In *Auto* or *FocusFollowsMouse* mode, the colormap is set to the colormap of the window below the pointer.
CirculateRaise	YES or NO	NO	Whether the window should be raised when circulating (focus the next or previous window through the keyboard).
AlignSubmenus	YES or NO	NO	Whether submenus should be aligned vertically with their parent menus.
OnTopTransients	YES or NO	YES	Whether transient windows should always be placed over their owner windows.
WindowPlacement	auto, cascade, manual or random	cascade	Selects placement style for new windows. *auto* places the window automatically in the first open space found in the workspace. *cascade* places the window in incrementing positions starting from the top-left corner of the workspace. *manual* allows you to place the window interactively with the mouse. *random* places the window randomly on the workspace.
OpaqueMove	YES or NO	NO	Whether the whole window should be moved while dragging it or if only it's frame should be moved.
NoAnimations	YES or NO	NO	Whether animations, like the one done during miniaturization, should be disabled.
NoAutowrap	YES or NO	NO	Do not automatically switch to the next or previous workspace when a window is dragged to the edge of

Appearance Options

Fonts are specified in the X Logical Font Description format. You can cut and paste these names from programs like `xfontsel`.

Colors are specified as color names in the standard X format. This can be any color name shown by the `showrgb` program (like black, white or gray) or a color value in the #rrggbb format, where rr, gg and bb is the intensity of the color component (like #ff0000 for pure red or #000080 for medium blue). Note that color names in the #rrggbb format must be enclosed with double quotes.

Textures are specified as an array, where the first element specifies the texture type followed by a variable number of arguments.

Valid texture types are:

(solid, color) the texture is a simple solid color.

(dgradient, color1, color2) the texture is a diagonal gradient rendered from the top-left corner to the bottom-right corner. The first argument (color1) is the color for the top-left corner and the second (color2) is for the bottom-right corner.

(hgradient, color1, color2) the texture is a horizontal gradient rendered from the left edge to the right edge. The first argument (color1) is the color for the left edge and the second (color2) is for the right edge.

(vgradient, color1, color2) the texture is a vertical gradient rendered from the top edge to the bottom edge. The first argument (color1) is the color for the top edge and the second (color2) is for the bottom edge.

Examples

(solid, gray)

(dgradient, gray80, gray20)

(hgradient, gray80, gray20)

(vgradient, gray80, gray20)

Option	Values	Default Value	Description
WorkspaceBack	texture	None	Default texture for the workspace background. Note: The *dgradient* texture can take a lot of time to be rendered.
IconBack	texture	(solid, gray)	Texture for the background of application icons and miniwindows.
FTitleBack	texture	(solid, black)	Texture for the focused window titlebar.
PTitleBack	texture	(solid, "#616161")	Texture for the unfocused main window titlebar or the owner of the currently focused transient window.
UTitleBack	texture	(solid, gray)	Texture for untitled window titlebars.
MenuTitleBack	texture	(solid, black)	Texture for menu titlebars.
MenuTextBack	texture	(solid, gray)	Texture for menu items.
FTitleColor	color	white	The color of the text in the focused window titlebar.
PTitleColor	color	white	Color for the text in the unfocused main window titlebar or the owner of the currently focused transient window.
UTitleColor	color	black	Color for the text in unfocused window titlebars.
MenuTitleColor	color	white	Color for the text in menu titlebars.
MenuTextColor	color	black	Color for the text in menu items.
HighlightColor	color	white	Color for the highlighted item in menus.
HighlightTextColor	color	black	Color for the highlighted item text in menus.
MenuDisabledColor	color	"#616161"	Color for the text of disabled menu items.
WindowTitleFont	font	helvetica bold 12	Font for the text in window titlebars.
MenuTitleFont	font	helvetica bold 12	Font for the text in menu titlebars.
MenuTextFont	font	helvetica medium 12	Font for the text in menu items.
IconTitleFont	font	helvetica medium 8	Font for the text in miniwindow titlebars.
DisplayFont	font	helvetica medium 12	Font for the text in information windows, like the size of windows during resize.
TitleJustify	center, left or right	center	Justification of the text in window titlebars.

Extra Appearance Options

Option	Values	Default Value	Description
MRightButtonBack	texture	(solid, black)	
FRightButtonBack	texture	(solid, black)	
PRightButtonBack	texture	(solid, "#616161")	
URightButtonBack	texture	(solid, gray)	
FLeftButtonBack	texture	(solid, black)	
PLeftButtonBack	texture	(solid, "#616161")	
ULeftButtonBack	texture	(solid, gray)	

Keyboard Bindings

Keyboard shortcut specifications are in the form:

[<modifier key names> +] <key name>

Where *modifier key names* specify an optional modifier key, like `Meta` or `Shift`. Any number of modifier keys might be specified. The *key name* is the actual key that will trigger the action bound to the option.

Examples:

F10 means the `F10` key.

Meta+Tab means the `Tab` key with the `Meta` modifier key pressed at the same time.

Meta+Shift+Tab means the `Tab` key with the `Meta` and `Shift` modifier keys pressed at the same time.

Key names can be found at `/usr/X11R6/include/X11/keysymdef.h` The `XK_` prefixes must be ignored (if key name is `XK_Return` use `Return`).

Option	Default Value	Description
RootMenuKey	None	Opens the root menu under the current pointer position and allows keyboard traversal of the menu. If the menu is already opened, it will allow keyboard traversal of it.
WindowListKey	None	Opens the window list menu under the current pointer position and allows keyboard traversal of the menu. If the menu is already opened, it will allow keyboard traversal of it.
WindowMenuKey	None	Opens the window commands menu for the current focused window and allows keyboard traversal of it.
MiniaturizeKey	None	Miniaturizes the current focused window if it's miniaturizable.
HideKey	None	Hides the current active application if it's hideable.
CloseKey	None	Closes the current focused window if it's closable.
MaximizeKey	None	Resizes the current focused window to it's full size.
VMaximizeKey	None	Resizes the current focused window to it's full vertical size.
RaiseKey	`Meta+Up`	Raises the current focused window to the top.
LowerKey	`Meta+Down`	Lowers the current focused window to the bottom.
RaiseLowerKey	None	Raises or lowers the window under the mouse pointer.
ShadeKey	None	Shades the current focused window.
FocusNextKey	None	Switches input focus to the next window.
FocusPrevKey	None	Switches input focus the the previous window.
NextWorkspaceKey	None	Switches to the next workspace.
PrevWorkspaceKey	None	Switches to the previous workspace.
NextWorkspaceLayerKey	None	Switches to the next 10 workspace set.
PrevWorkspaceLayerKey	None	Switches to the previous 10 workspace set.
Workspace1Key... Workspace10Key	None	Switches to the workspace with the corresponding number.

4.1.3 Window Attributes

Window attributes are stored in the `~/GNUstep/Defaults/WMWindowAttributes` file.

Syntax

The contents of this file is a dictionary of attribute dictionaries keyed by window names. Like this:

```
{
        "*" = {
                Icon = "defaultAppIcon.xpm";
        };
        "xterm.XTerm" = {
                Icon = "xterm.xpm";
        };
        xconsole = {
                Omnipresent = YES;
                NoTitlebar = YES;
                KeepOnTop = NO;
        };
}
```

Window names are in the form: [1]

<window instance name> . <window class name>
OR
<window instance name>
OR
<window class name>

Placing an asterisk as the window name means that the values set for that key are to be used as default values for all windows. So, since xconsole does not specify an `Icon` attribute, it will use the default value, which in the above example is `defaultAppIcon.xpm`.

Options

Default values are NO for all options.

[1] You can get the values for these information by running the `xprop` utility on the desired window. When you do that, it will show the following line, among other things:
`WM_CLASS(STRING) = "xterm", "XTerm"`
The first string (`xterm`) is the window instance name and the second (`XTerm`) the window class name.

Option	Values	Description
Icon	pixmap file name	Assigns a pixmap image to be used as the icon for that window.
NoTitlebar	YES or NO	Do not put a titlebar in the window.
NoResizebar	YES or NO	Do not put a resizebar in the window.
NotMiniaturizable	YES or NO	Do not let the window be miniaturized and remove the corresponding button from the titlebar.
NotClosable	YES or NO	Remove the close button from the titlebar.
NoHideOthers	YES or NO	Do not hide the window or the application to which the window belongs when a —Hide Others— command is issued.
NoMouseBindings	YES or NO	Do not grab mouse buttons in that window. This means that actions like `Meta` Click on the window will be caught by the application instead of being intercepted by WindowMaker.
NoKeyBindings	YES or NO	Do not grab keys in that window. This means that keystrokes that would be normally intercepted by WindowMaker (because they are bound to some action), like `Meta` + `Up`, will be passed to the application.
NoAppIcon	YES or NO	Do not create an application icon for the window. This is usefull for some applications that incorrectly get more than one application icon.
KeepOnTop	YES or NO	Always keep the window over other normal windows.
Omnipresent	YES or NO	Make the window be present in all workspaces. AKA sticky window.
SkipWindowList	YES or NO	Do not list the window in the window list menu.
KeepInsideScreen	YES or NO	Always keep window inside the visible area of the screen.
Unfocusable	YES or NO	Do not let the window be focused.
StartWorkspace	workspace number or name	Make the window be shown in the indicated workspace when it's first shown.

4.1.4 Applications or Root Menu

Menu File Syntax

4.2 Configuration Panels

4.2.1 Preferences Panel

4.2.2 Window Attributes Inspector

Appendix A

Tips

- If the size of a window gets so large that it doesn't fit on the screen and you can't manipulate it, you can simply hold the `Meta` key while dragging the window in the client area. This way you can move the window up or down and resize it, if you want.
- If you want windows to be able to cover the dock, you can make the dock lowerable by double clicking the first dock icon while holding the `Meta` key. Then, you can raise and lower the dock through the first icon, just like you do with windows.
- If you want windows to be able to cover menus, you can make them lowerable just like the dock by double clicking the titlebar with the `Meta` key pressed.

Appendix B

Glossary

drag to click in an object with the mouse and move the mouse while holding the mouse button.

miniaturize (iconify, minimize) to temporarily put a window aside, replacing the window with a miniature representation of it.

Meta key depending on the system and keyboard types, this can mean different keys. Under Linux, it is usually the `Alt` or `Alternate` key.