# Table of Contents

Preface ................................................................................. 1
   How to use this manual ................................................... 1
   Typographical Conventions ............................................. 1
   Acknowledgments .......................................................... 1

1 What is X? ........................................................................... 3
   System components ......................................................... 3
      Servers and clients ........................................................ 3
      Resource database ...................................................... 4

2 Getting started ................................................................. 5
   The default environment using GNOME ............................ 5
      What is GNOME? ........................................................ 5
         The initial screen configuration .................................. 6
         Working with GNOME .............................................. 8
      Using the K Desktop Environment ................................. 10
         GNOME vs. KDE ..................................................... 10
         Switching to KDE .................................................. 11
         The initial screen configuration ................................ 11
         Working with KDE ................................................. 14
      The default environment using mwm .............................. 15
         The initial screen configuration ................................ 15
         Using the menus .................................................... 17
      Working in the default environment ............................... 18
         Moving windows .................................................... 19
         Resizing windows ................................................... 19
         Creating new xterm windows ..................................... 19
         Iconifying and de-iconifying windows ....................... 21
         Raising windows .................................................... 22
         Scrolling .................................................................. 22
         Cut and paste .......................................................... 22

3 Customizing your environment ........................................... 25
   Customizing your GNOME desktop environment ............... 25
      Using the GNOME Control Center ................................ 25
      Using Context Menus ............................................... 26
   Customizing your K desktop environment .......................... 28
      Customizing the K Menu ............................................ 30
   Customizing your X environment ..................................... 31
Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanisms for changing the default environment</td>
<td>32</td>
</tr>
<tr>
<td>Customization examples</td>
<td>32</td>
</tr>
<tr>
<td>Making changes at the command line</td>
<td>32</td>
</tr>
<tr>
<td>Changing window size and/or placement</td>
<td>33</td>
</tr>
<tr>
<td>Background image for the root window</td>
<td>34</td>
</tr>
<tr>
<td>Starting applications from your .xinitrc file</td>
<td>35</td>
</tr>
<tr>
<td>Running X clients from foreign hosts</td>
<td>35</td>
</tr>
<tr>
<td>4 Advanced customization</td>
<td>37</td>
</tr>
<tr>
<td>Specifying resources</td>
<td>37</td>
</tr>
<tr>
<td>Specifying class names of applications</td>
<td>38</td>
</tr>
<tr>
<td>Using instance names of applications</td>
<td>38</td>
</tr>
<tr>
<td>Applying resources to all windows</td>
<td>39</td>
</tr>
<tr>
<td>Class and instance names of resource options</td>
<td>40</td>
</tr>
<tr>
<td>Modifying the window manager, mwm</td>
<td>40</td>
</tr>
<tr>
<td>Using resources</td>
<td>40</td>
</tr>
<tr>
<td>Using the resource description file: .mwmrc</td>
<td>42</td>
</tr>
<tr>
<td>Changing window text font</td>
<td>45</td>
</tr>
<tr>
<td>Listing available fonts</td>
<td>45</td>
</tr>
<tr>
<td>Viewing fonts</td>
<td>46</td>
</tr>
<tr>
<td>Specifying fonts</td>
<td>46</td>
</tr>
<tr>
<td>Adding font paths</td>
<td>46</td>
</tr>
<tr>
<td>Glossary</td>
<td>49</td>
</tr>
<tr>
<td>X utilities</td>
<td>51</td>
</tr>
<tr>
<td>Miscellaneous Utilities</td>
<td>51</td>
</tr>
<tr>
<td>Adjusting your X environment</td>
<td>52</td>
</tr>
<tr>
<td>Cut buffer and clipboard</td>
<td>52</td>
</tr>
<tr>
<td>X Window &quot;dumps&quot;</td>
<td>53</td>
</tr>
<tr>
<td>X Status Utilities</td>
<td>53</td>
</tr>
<tr>
<td>X Resource utilities</td>
<td>54</td>
</tr>
<tr>
<td>Font Utilities and Conversion programs</td>
<td>54</td>
</tr>
<tr>
<td>Bitmap utilities</td>
<td>55</td>
</tr>
<tr>
<td>FIG</td>
<td>55</td>
</tr>
<tr>
<td>X interfaces to other programs</td>
<td>55</td>
</tr>
<tr>
<td>Display/demo programs</td>
<td>55</td>
</tr>
<tr>
<td>References</td>
<td>57</td>
</tr>
</tbody>
</table>
Preface

How to use this manual

This document is an introduction to the X Version 11 Window System, Release 5 and 6, the GNOME and KDE desktop environments used in LINUX, and the Motif Window Manager. We have designed this manual for the School of Computer Science community and have therefore compiled it with an experienced computer user in mind. If you’re new to our computing environment, we suggest you take a look at the Introduction to the SCS Computing Environment [3], available in WeH 5215 or online in the Web version at the following URL:

http://www.cs.cmu.edu/~help/introdotscs_intro

Typographical Conventions

We use square brackets to indicate special keys, such as [BREAK] and [ESCAPE]. Key names may also differ on various machines. For example, some keyboards do not have [ALT]. If you have a PMAX, you should use [COMPOSE CHARACTER] in place of [ALT]. On the Sun-3, use the keys labeled [RIGHT] or [LEFT].

Command lines and program names are in boldface type. When you see bold italics, fill in a particular value for the italicized item; for example, ls directoryname might become ls letters.

In this manual we use the three-button mouse as standard. If you have a two-button mouse, press both buttons simultaneously to simulate pressing a middle button.

Acknowledgments

We thank the SCS Research Systems and Computing Facilities staffs for their expertise and assistance in reviewing this manual. Please direct comments and suggestions about this document to the Help Desk (“help@cs”).
1 What is X?

X is a “windowing” system developed at MIT. Windows subdivide the display screen; each represents an input/output stream between processor and display, and each may be used to run a particular program. Windows may overlap in a stack, and you may move from window to window as needed. Thus, you can use several different programs simultaneously and see the output on your screen in separate windows as if each were a separate terminal.

X supports text and graphics operations on both monochrome and color displays. X is also device-independent: Applications written for X can be displayed on any workstation that runs X. See the UNIX manual entry (type man X) for general information on X.

System components

Servers and clients

X comprises a server and one or more client processes. A server controls access to a particular display and monitors your use of the keyboard and mouse. Typically, your workstation has only one display and runs only one server. Client programs connect to the server for access to both the display and any keyboard and mouse activity that is meant for the client. For example, if you start a client called xcalc (a calculator), it connects to your server to display a calculator on your screen. If you move your mouse so that the cursor is over the calculator, the server sends any mouse clicks or keystrokes to xcalc as input.

In the standard SCS environment, an X server for your display is started from your .login file. Several X clients are also started automatically: a window manager, a terminal emulator, and a clock. The first two are described below along with a third that is started if you are using a workstation running LINUX.

The window manager

The window manager provides functions that perform operations on windows and icons such as raising, lowering, or moving. If you run X without also running the window manager client, you could have multiple windows; however, you would not be able to move the
windows, or “raise” windows obscured by other windows. The default window manager supported by SCS Computing Facilities is \texttt{mwm}, the Motif Window Manager. \texttt{twm}, the Tab Window Manager, is also available; to run it, remove the “#” from the line “\#exec twm > \hspace{1em} /dev/null” in your .xinitrc file and put it at the front of the line “exec mwm>/dev/null”. See \texttt{man twm} for more information.

Other volunteer-supported window managers (for example, \texttt{fvwm} or \texttt{ctwm}) may be available, but they are not officially supported.

\textbf{Terminal emulators}

When \texttt{X} starts it creates a “root window” on your display screen. The SCS default profiles then start a terminal emulator called \texttt{xterm} that allows you to communicate with the operating system. The emulator simulates the behavior of some type of terminal, such as a Concept or a VT100.

\textbf{Desktop environments}

When \texttt{X} starts on a workstation running a facilitated installation of Red Hat Linux 7.1 or greater, the GNOME desktop environment will start by default. This “environment” provides for a drag-and-drop capable desktop, a panel that allows you to start programs and display their status and/or output, and a standardized “look and feel” for applications that comply with the standards of the desktop environment.

\textbf{Resource database}

When you start an \texttt{X} server, it reads the .xinitrc file in your home directory. Typically, the first command in this file runs \texttt{xrdb}, which creates or modifies the \texttt{resource database}, a data structure internal to the server. This database records your preferences for client programs. These preferences, called \texttt{resources}, specify how an \texttt{X} program should look or behave. For example, you can make sure that \texttt{xclock} brings up an analog clock instead of a digital clock whenever you invoke it by specifying the \texttt{xclock} resource “analog” in your .Xdefaults file. Resources stored in a server are available for all clients connected to that server. See page 37 for more on resources and how to specify them.
2 Getting started

If you are a new workstation user, simply log in and X will automatically start. Your home directory contains profile files that initialize your X environment.

If you have a nonstandard ".login" but want to compare it with how the standard login invokes X, we recommend that you look at the following SCS default profile files:

/usr/local/lib/profiles/login
/usr/local/lib/profiles/cshrc

If you start X but don’t have the files .xinitrc, .Xdefaults, or .mwmrc, they will be created for you. Note, however, that if you do have these files, the system will not automatically replace them with the standard ones. The standard profiles have the following paths:

/usr/local/lib/profiles/xinitrc
/usr/local/lib/profiles/Xdefaults
/usr/local/lib/profiles/mwmrc

For more information on profile files see the discussion starting on page 25.

The default environment using GNOME

What is GNOME?

 GNOME stands for GNU Network Object Model Environment. It is a desktop environment that helps users configure and use their workstations via a graphical interface. It consists of a number of components:

• The Panel: a combination status bar and application launcher.
• The Desktop: a drag and drop capable background for organizing shortcuts.
• Tools and Applications: a suite of utilities that provide a graphical interface for common applications and customization utilities.

 GNOME also includes a set of standards used by programmers to help make the software on your machine cooperate and communicate more easily and effectively with each other.

The initial screen configuration

The illustration below shows a workstation screen running GNOME and the sawfish window manager with the standard configuration. There are two main components to this configuration: the desktop and the panel.

The desktop provides not only a background for your screen but also serves a number of functions that help you to keep your workstation organized and efficient. Due to its drag-and-drop capabilities, the desktop can hold launchers which serve as shortcuts to your most important directories, internet URLs, and applications. The default desktop provides access to your home directory, your floppy drive, your CD-ROM drive, the Red Hat web site and a trash can used to delete unwanted files and shortcuts. To use these launchers, simply double click on them with the left mouse button and the item that the launcher refers to will open.

The desktop also provides you with a number of workspaces that you can use to organize your work. Each workspace is a separate desktop where you can store the various windows that you are using. This feature allows you to create customized workspaces that contain only the windows necessary to complete a given task, without windows
from other tasks cluttering your view. You can change between workspaces by clicking on the boxes located in the center of the panel. Each box will take you to a different workspace.

The Panel
The panel is a strip that, by default, sits at the bottom of your desktop. It can hold a number of different components that can make interacting with your workstation faster and easier. The panel is drag-and-drop compliant and is capable of holding a number of different objects, including menus, launchers, applets, and some other special objects. As the principle component of interaction with a GNOME based system, it is important to become familiar with the components of the panel and their various capabilities.

Panel Objects
The panel is capable of housing a wide variety of objects that allow you to perform numerous tasks including: launching applications, managing windows and workstations, searching for help and gathering information about the performance of your workstation. These are all accomplished through panel objects which sit inside the panel. There are five principle types of panel objects that perform specific tasks. They are:

- **Menus**: These objects, similar to the start menu in Windows operating systems and the apple menu in Macintosh operating systems present a list of program launchers, special items, applets, and submenus. Menu objects make the task of finding applications and applets easier and faster by allowing you to organize them quickly and simply.

- **Launchers**: Program launchers are panel objects that start a specified program when you click on them. Program launchers can be customized to launch applications with different options enabled or disabled. You can also specify the icon that you would like the launcher to display.

- **Applets**: Applets are special applications made to run within the panel. They can either provide information, entertain, or enhance interactivity. Like all objects in the panel, they can be customized.

- **Drawers**: Drawers are basically panel extensions that allow you to store more objects in the panel than the panel can visibly handle. Drawers, by default, have an icon that looks like a chest of drawers (although you are welcome to change the icon to anything that you like). As extensions of the panel, you can store any object that can operate within the panel inside a
drawer. Drawers also can help to organize the panel and make it possible to keep your on-screen workspace simple and organized.

- **Special Objects**: There are a number of special objects that can live in the panel that allow users to perform specific tasks. For example, they can allow you to run programs from a command line dialog box, they can initiate your xScreensaver, or they can log you out of an X-Windows session.

**Windows**

The window manager, **sawfish** by default, encapsulates each window inside a frame that contains a number of interactive objects, known as **widgets** that allow you to control the behavior of the window.

The top of the window is called the **titlebar** and contains the **window menu** button on the left end and three control buttons, similar to those used in Microsoft Windows operating systems to the right that control window positioning. From left to right, they are the **minimize** button, the **maximize** button, and the **close** button. The space in between the menu button and the three control buttons is known as the **title area**. Around all sides and on all four corners of the window is a **border** that allows you to resize the window. Moving the mouse to a border will change your pointer to a **resize arrow**. There is also a **scroll bar** running down the right side of the window, just inside the border.

Only the active window can receive input from the keyboard and the mouse. Under the default GNOME environment, the title area of the active window is highlighted in a different color than the title area of all other windows on the screen. In order to interact with a window, you should click in it once to activate it.

**Working with GNOME**

**Using menus**

GNOME uses a system of menus to help you control and configure your machine. The most important of these is the **Main Menu**, which in the default environment is the rightmost object in the panel, represented by the RedHat icon. The main menu behaves much like the “start” button in Microsoft Windows operating systems, presenting you with quick access to programs, internet shortcuts, applets, and other tools that allow you to run commands, configure your screen, and log out of the system.
The various submenus within the main menu help to maintain organization among the launchers that are available in the default environment. Initially, you will find the following submenus:

- **Accessories, Games, Graphics, Internet, Office, Programming, Sound & Video, and System Tools**: which provide access to most of the applications available on your system that have a graphical interface.
- **Preferences**: which gives you fast access to a number of user configuration options.
- **System Settings**: which provides an interface for making system-wide configuration changes. You must be root to use these.
- **System Tools**: which provides access to a number of applications that simplify the administration of your workstation.

In addition to these submenus, the main menu gives you fast access to a number of other tools and functions:

- **Help**: starts a help browser with information about GNOME.
- **Home Folder**: displays the contents of your home directory.
- **Network Servers**: provides access to a number of MS Windows servers on the CS network.
- **Run...**: opens a single line command prompt, allowing you to enter commands as you would from inside a terminal window.
- **Search for Files...**: opens a utility for finding files.
- **Open Recent**: provides access to recently accessed documents.
- **Lock Screen**: This special menu object starts an xscreensaver which locks your screen. Currently, this feature is not supported in our environment and therefore, you will not be able to unlock your screen with the same userid and password combination that you use to log into your workstation.
- **Log out**: is another special menu object that will log you out of your X session. This, in turn, will log you out of your workstation unless you uncomment the set_no_X variable in your .login file. Doing this will drop you to a console prompt. In this case, you will need to type `logout` to end your session.

In addition to the main menu, each window in GNOME contains a menu. In the default environment, in which GNOME works together with a window manager called sawfish, there will be a window menu button at the upper left corner of every window that has a title bar. Left-clicking on this button displays the window menu. This menu offers the following choices:
• **Minimize**: removes the window from the desktop. You can retrieve it by clicking on the bar displayed in the center of your panel that corresponds to the minimized window. When minimized, the icon on the left side of this bar will appear to be “greyed-out”.

• **Maximize**: increases the size of the window to fit the visible size of your display. Only the maximized window and the panel will be visible. You can return the window to normal size by selecting **Unmaximize** from the menu or by clicking the unmaximize button which is the second from the right in the menu bar of any maximized window.

• **Roll Up**: hides the contents of a window, displaying only the title bar.

• **Move**: positions the window according to the position of the cursor.

• **Resize**: resizes the window.

• **Close**: closes the window. If the window is the only or final remaining window for an application, that application will exit.

• **Put on all workspace**: makes the window appear on all virtual desktops.

• **Move to...**: places the window on a new desktop.

In addition to the main and window menus, there are other menus that are available to you directly from the GNOME desktop. Clicking the right mouse button on the desktop displays a menu that allows you to configure the appearance of the desktop. From here you can create new launchers, organize your desktop and even change the background image.

### Using the K Desktop Environment

Although it is not the default desktop environment on facilitized SCS Linux machines, the **K Desktop Environment (KDE)** is available for your use. This environment serves many of the same functions as the GNOME desktop environment. Like GNOME, its purpose is to provide a simple and consistent interface for the configuration and use of your workstation.

### GNOME vs. KDE

The question of why to choose one desktop environment over another is largely a matter of personal preference. Since many programs are developed using toolkits which correspond directly to GNOME or KDE, you may find it useful to choose the desktop environment that is most compatible with the software you use most frequently.
Switching to KDE

Although the default desktop environment in facilitized installations of Red Hat Linux 7.1 is GNOME, KDE is accessible through the use of the `switchdesk` command. With X running, type `switchdesk` and hit [enter] to bring up a graphical dialog box with a menu of the available desktop environments on your machine. If you know the name of the desktop environment you want, you can specify it directly by typing the name of the environment after the command. For example, to switch to KDE, type:

`switchdesk kde`

The change will occur the next time you start X.

The initial screen configuration

When KDE starts, the initial screen configuration consists of two primary components: the desktop and the Kicker.

The Desktop

KDE is capable of displaying several desktops which can hold links to frequently accessed applications and documents. The desktop is drag-and-drop compliant and can accept links from almost any drag-and-drop compliant application. Therefore, by simply dragging the icons for documents and URLs to the desktop you can organize your workspace to provide fast access to the tools and information you use most. The default desktop provides access to your Autostart folder (see chapter 3), your floppy drive, your CD-ROM drive, your default printer,
the Red Hat web site, and Linux documentation, as well as a trash can used to delete unwanted files and shortcuts. To use these shortcuts, simply double click on them with the left mouse button and the item that the shortcut refers to will open.

The Kicker

At the bottom of the desktop is a fully-configurable panel which provides fast access to a number of applications, configuration tools, and system information called the Kicker. While the Kicker is capable of holding a diverse collection of items, the default configuration contains the following from left to right:

- **Application Starter (K) Menu**: which provides access to the most common applications and configuration tools. It is opened by clicking on the K icon located in the leftmost position on the panel.
- **Show Desktop button**: minimizes every window on your screen to expose the desktop behind them.
- **Terminal button**: which opens a terminal running your default shell.
- **Control Center button**: which starts the K Control Center, a configuration tool that provides access to a wide variety of customization options offered by the K Desktop Environment.
- **Help button**: this icon launches the Konqueror web, document, and file system browser with K help documentation loaded.
- **Home Directory button**: which launches Konquerer with your home directory contents loaded.
- **Konqueror Browser button**: launches the Konqueror browser, used to view documents, web pages, and as a gui front-end to make navigating through the file system easier.
- **Application buttons**: which launch various useful programs. By default the SCS environment provides the following: Kword word processor, KSpread spreadsheet, KPresenter presentation editor, Killustrator drawing tool, and Advanced Editor text editor.
- **Desktop buttons**: next are the desktop buttons which allow you to navigate between different virtual desktops or workspaces.
- **Taskbar**: which contains buttons that allow you to minimize and maximize windows.
- **Lock button**: which locks your screen (this is currently unsupported).
- **Logout button**: located directly under the lock button, this logs you out of the K desktop environment and if you have X set to start automatically upon login, will log you out of the workstation.
- **Klipper button**: This button gives you access to the items most recently added to the clipboard.
• System clock

Kicker Objects

The Kicker is capable of hosting the following two types of items:

• **Buttons**: which serve as links to other programs, documents, and URLs. They can also perform special functions like clearing your desktop or ending your X session.
• **Applets**: which are small programs that display output in the Kicker.

Windows

As with any X window manager, KDE encapsulates each window inside a frame that contains a number of interactive objects, known as *widgets* that allow you to control the behavior of the window.

![KDE Window](image)

The top of the window is called the *titlebar* and contains the *window manipulation icon* on the left end followed by the *sticky button* and three control buttons, similar to those used in Microsoft Windows operating systems to the right that control window positioning. From left to right, they are the *iconify button*, the *maximize button*, and the *close button*. The space in between the menu button and the three control buttons is known as the *title area*. Around all sides and on all four corners of the window is a *border* that allows you to resize the window. Moving the mouse to a border will change your pointer to a *resize arrow*. There is also a *scroll bar* running down the right side of the window, just inside the border.
Only the active window can receive input from the keyboard and the mouse. Under the K desktop environment, the title area of the active window is highlighted in a different color than the title area of all other windows on the screen. In order to interact with a window, you should click in it once to activate it.

**Working with KDE**

**Using menus**

KDE gives you control over your machine and its configurable options through a series of menus. The principle of these is the Application Starter (or K Menu), which in the default environment is the leftmost object in the panel, represented by a K and a gear. This menu offers access to programs, internet shortcuts, applets, and other tools that allow you to run commands and log out of the system.

The various submenus within the main menu help to maintain organization among the launchers that are available in the default environment. Initially, you will find the following submenus:

- **Accessories, Games, Graphics, Internet, Office, Programming, Sound & Video**: a collection of assorted programs.
- **Preferences**: a shortcut to the K Control Center configuration tool.
- **System Settings**: system information and control utilities (requires root access).
- **System Tools**: system administration tools.

In addition to these submenus, the Application Starter gives you fast access to a number of other tools and functions:

- **Control Center**: KDE’s main configuration utility.
- **Find Files**
- **Help**
- **Home Directory**: opens a Konqueror browser to your home directory.
- **Run Command...**: provides a command line prompt to run programs manually.
- **Lock Screen**: locks your screen with a screensaver. You must set a local password to use this.
- **Logout**: will end your X session and if you have X run automatically when you log in, will log you out of the workstation.
In addition to the Application Starter, each window in KDE has a window manipulation icon that opens a window menu. This menu offers the following choices:

- **Move:** which allows you to change the position of the window on the desktop.
- **Size:** which allows you to make the window smaller or larger.
- **Minimize:** which hides the window, leaving an icon in the Kicker.
- **Maximize:** which expands a window to match the size of the desktop.
- **Shade:** which “rolls up” the window, leaving only the title bar.
- **Always on Top:** this selection will keep the window on top, even when it does not have the focus.
- **Store Settings:** sets the current settings as default for this window.
- **Configure Window Behavior:** allows you to change the look of all windows displayed by KDE.
- **To Desktop:** sends the window to a different desktop (or workspace).
- **Close:** closes the window.

Middle or right-clicking on any empty region of the desktop provides context menus that offer quick access to a variety of features in KDE. Middle-clicking opens a context menu that will bring the focus to any window in any desktop. Right-clicking displays a menu offering access to bookmarks, clipboard functions, and a number of configuration tools. From here you can create new shortcuts, organize your desktop and even change the background image.

### The default environment using mwm

#### The initial screen configuration

The figure below shows a workstation screen with the standard configuration. The *root window* is the dark gray area behind the other windows. Three other windows appear in the root window:

- A large window on the left created by **xterm**. It contains your UNIX prompt and functions as a terminal screen.
- A digital clock in the upper right corner.
- The icon manager, which lists all windows for clients currently connected to the server (initially `xclock` and `xterm`). The icon manager allows you to see all current windows in one place, whether or not they are iconified. See page 21 for more information on the icon manager.

The window manager, `mwm`, surrounds each window with a frame containing several features known collectively as widgets:

- At the top of the `xterm` window: The titlebar.
  - On the left end of the titlebar: The window menu button.
  - On the far right end: The maximize, or full-zoom, button.
  - Next to the maximize button: A button with a tiny square in it called the minimize, or iconify, button.
  - The area between the buttons and containing the window title: The title area.

- Around the window: A border in eight sections, four long edges and four corners. Resize pointers appear when you move the mouse to the edge of the window.

- Down the left side of the window: The scroll bar.

The section starting on page 18 explains how to use these widgets.
A window may take input from the keyboard/mouse and then display output, as do `xterm` or `xcalc` windows, or it may display only program output, as in the clock window. When you put the mouse pointer in a window and click the left button, that window becomes the “active” or “listener” window and receives all keyboard input. An active window, like the one labeled “ladon.resdoc” in the illustration, has the following characteristics:

- The text cursor, if it exists, is highlighted.
- The window frame is highlighted.
- Keyboard and mouse input goes to the window.

### Using the menus

The standard X environment includes two window manager menus: The Window Menu, which contains items that affect the appearance of individual windows; and the Root Menu, which contains items that affect the appearance of the entire screen. The Window Menu can only be raised from within a window; your menu choice will then affect the window from which it was raised. The Root Menu can only be raised from the root window and contains items that affect all windows.

To get the Window Menu:

- Put the mouse pointer in the window menu button (the button at the left end of the title bar) of the window you want to affect. Click the left mouse button to pop up the menu. Then click on your selection (or click outside the menu to make it go away).
- You can also put the pointer in the title bar of the window you want to affect. Press the middle mouse button to bring up the menu and drag the pointer to your selection. Release the button.

or, if the window has no window menu button:

- Click the mouse pointer in the window you want to affect.
- Hold down [ALT] and hit the [SPACE] bar.
- Click on your choice (or click outside the menu to make it go away).

[ALT] is the same as:

- [COMPOSECHARACTER] on a Pmax
- [RIGHT] or [LEFT] on a Sun
See the following table for descriptions of options.

<table>
<thead>
<tr>
<th>The Window Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restore size</strong>—Restore a full-zoomed window to its previous size.</td>
</tr>
<tr>
<td><strong>Move window</strong>—Move a window. After you click on this item, position the window with the mouse; then click the left button to place the window.</td>
</tr>
<tr>
<td><strong>Resize window</strong>—Change window size. After you select this item, the mouse pointer changes to a cross. Move the cross over any border and click the left mouse button when the size is correct.</td>
</tr>
<tr>
<td><strong>Iconify window</strong>—Convert the window to an icon.</td>
</tr>
<tr>
<td><strong>Full-zoom</strong>—Enlarge the window to its maximum size (usually the whole screen).</td>
</tr>
<tr>
<td><strong>Lower window</strong>—Put the window on the bottom of the stack.</td>
</tr>
<tr>
<td><strong>Kill window</strong>—Kill the client process and remove the window.</td>
</tr>
</tbody>
</table>

To get the Root Menu:

- Put the mouse pointer in the root window and press the left mouse button.
- Drag the pointer to your selection and release the button. If you don’t want to choose a menu item, move the pointer off the menu and release the button.

See the following table for descriptions of options.

<table>
<thead>
<tr>
<th>The Root Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New window</strong>—Spawn a new shell process and open a window to it.</td>
</tr>
<tr>
<td><strong>Circle up</strong>—Raise the bottom window to the top of the stack.</td>
</tr>
<tr>
<td><strong>Circle down</strong>—Lower the top window to the bottom of the stack.</td>
</tr>
<tr>
<td><strong>Refresh screen</strong>—Redraw the entire screen.</td>
</tr>
<tr>
<td><strong>Restart mwm</strong>—Terminate and restart the window manager.</td>
</tr>
</tbody>
</table>

**Working in the default environment**

In this section we describe how to perform basic tasks in the X environment. Note that some tasks can be performed with the mouse, the keyboard, or the menus described earlier.
Moving windows

To move a window, choose the “Move window” option of the Window Menu (see page 17), or:

- Put the mouse pointer over the title area and press the left mouse button.
- Drag the window by moving the mouse. As you move the mouse, an outline of the window follows the mouse movement to indicate the new location. You’ll also see a small box that describes the coordinates (+X,+Y) of the new position.
- Release the mouse button, and the window will be relocated.

Resizing windows

To resize a window, choose the “Resize window” option in the Window Menu (see page 17), or:

- Put the mouse pointer over a window border. The pointer will change to a bar and arrow.
- Press the left mouse button and drag the edge to the new location. As you resize the window, a box appears on the screen describing the size (WxH) of the window as you move it.
- Release the button.

To move both the horizontal and vertical edges simultaneously, move the mouse pointer into a corner. The pointer will change to an arrow pointing at the corner. Move the arrow until the window is the size you want.

Creating new xterm windows

To your workstation

There are two ways to create a new xterm window for your workstation:

- Choose the “New Window” option of the Root Menu (see page 18).
- Type xterm & (The ampersand puts the process in the background, allowing you to use the original window as well.)
The window acts like a new terminal connected to your workstation, as illustrated below, where the user is editing a file in one window and reading a **man** page in the other.

To a different machine

If you have accounts on several machines, you can provide a window for each using the **xtelnet** alias provided in your .cshrc. First, remove the comment from the xtelnet line in your .cshrc. Then, if you want to work on another machine, for example, UX1.SPCS.CMU.EDU, type

```
xtelnet ux1
```
A window grid appears. Use the mouse to position the window and click the left mouse button to establish the window. The new window emulates a terminal with a telnet connection to the machine you specify, as illustrated below. Log in just as at a terminal.

Iconifying and de-iconifying windows

The icon manager lists all windows currently controlled by the window manager. When a window is created, an entry for that window appears on the list under the icon manager. When you “iconify” a window, the window disappears and its entry in the icon manager list is highlighted. Iconifying windows can reduce screen clutter when you don’t need a certain window for a while or if you have a long job running in a window. Note that processes in an iconified window continue to run.

To iconify a window, choose the “Iconify window” option in the Window Menu (see page 18), or:

- Move the mouse pointer to the minimize button, the second button from the right end of the title bar.
- Click with the left mouse button.
To de-iconify a window, move the mouse pointer to the window name in the icon manager’s list. Double-click the left mouse button.

The name “icon manager” is somewhat misleading: Instead of becoming conventional icons, windows become entries in the “Icons” window. If you prefer to have “real” icons, you can disable the icon manager by placing commenting characters (two exclamation marks) at the beginning of this resource option in your .Xdefaults file:

```
Mwm*useIconBox:True
```

### Raising windows

If you have several windows on your screen, one may partially or fully obscure another, overlapping as if they were physically stacked.

- To raise a window, move the mouse pointer into the window and click the left mouse button.
- To raise the next window, press [ALT], then type [TAB].

If you’d prefer to have the pointer focus on and raise to the top whatever window it’s in, see “keyboardFocusPolicy” and “focusAutoRaise” on page 40 and page 41.

### Scrolling

You have the option of saving the text that has scrolled off screen. The standard environment provides a scrollbar, which is the grey strip running the height of your **xterm** window on the left-hand side. The default number of lines saved is 64, determined by the resource `saveLines` (see page 37 for information on how to specify resources). To see text that has scrolled off screen:

- Place the pointer in the scrollbar. If only part of the scrollbar is highlighted, place the pointer in the gray area.
- Click the right mouse button to scroll the text down.
- Click the left mouse button to scroll the text up.
- Click the middle mouse button to indicate a point in the file with the pointer (position along the scrollbar corresponds to position in the saved text). A screenful of text from the arrow’s location in the text will be displayed.
- Press and hold the middle mouse button, moving the mouse pointer up and down along the scroll bar to scroll the text at your own pace.
- Hit any key to return to your prompt.

### Cut and paste

To reuse command lines or to copy text from one text window to another, you can use the “cut and paste” mechanism.
• Define the region of text you want to copy:
  • Place the pointer at the beginning of the targeted text and press the left mouse button.
  • Without releasing the button, drag the pointer down the screen until you reach the end of the text you wish to define or “cut”. Release the mouse button.
  • Move the pointer to where you want to insert the text and “paste” the defined region of text by clicking the middle button.

You can increase or decrease the size of the region after it’s already defined by pressing the right button and further dragging the mouse. To deselect a region, click the left mouse button outside the highlighted region. Quick double or triple clicks will select a word or line.

The region of text that you have defined or cut is stored in the “cut buffer.” Because only one segment of text is stored in the cut buffer and remains there until it is overwritten by a new piece of text, you can reuse the cut buffer contents many times. If you define too large a region, the latter portion might be truncated.
3 Customizing your environment

There are a number of ways that you can customize your working environment to suit your needs whether you use GNOME, KDE, or X alone. This chapter describes a number of basic modifications that you can make quickly and easily. Changes in GNOME and KDE are generally made through a graphical interface, while changes to X environments are usually made via configuration file modifications.

Customizing your GNOME desktop environment

There are three principal components of the GNOME desktop environment that you can alter to suit your needs: the desktop, the panel, and the main menu. Most of these changes can be made from one of two locations: the GNOME control center and the context menu that appears when you right click on one of the components.

Using the GNOME Control Center

Most configuration changes for GNOME and for the sawfish window manager can be made through the Main Menu. You can find the relevant control panels under the preferences submenu. The various categories of configurable items are:

- **About Myself**: which customizes personal information.
- **Accessibility**: which enables features that assist physically challenged users.
- **Background**: which changes the background image.
- **CD Properties**: which customizes the system’s reaction to the insertion of cds and dvds.
- **File Management**: which alters icon look and behavior.
- **File Types and Programs**: which allows you to associate file extensions with applications.
- **Font**: which configures default display fonts.
- **Handheld PDA**: which configures Palm Pilot syncronization.
- **Keyboard**: which alters keyboard behavior.
- **Keyboard Shortcuts**: which allows the creation of time-saving hotkeys that perform actions.
- **Login Photo**: is disabled, because gdm is not supported in our environment.
• **Menu and Toolbar:** which changes the look and feel of menus and toolbars.

• **Mouse:** which alters mouse behavior.

• **Network Proxy:** which allows users to direct network traffic from their workstation through a proxy. A direct connection is the default supported configuration.

• **Passwords:** is disabled in the SCS environment.

• **Preferred Applications:** which allows you to specify programs you would like to run by default for certain tasks.

• **Screensaver:** which configures screensaver module options.

• **Sound:** which controls audio preferences.

• **Theme:** which controls the appearance of windows, dialog boxes and warning messages.

• **Windows:** which configures mouse focus over windows.

More control panels for manipulating the sawfish window manager are available under **More Preferences**. To configure any item, simply choose it from the menu and select the options that you prefer from the choices that appear. **Close** exits the panel, applies your changes and makes them the default for future sessions.

**Using Context Menus**

You can also make changes to the configuration of your system through the series of context menus that appear when you right click on the desktop, the panel, and on desktop and panel objects.

**Desktop Context Menus**

By right-clicking on the desktop, it is possible to add new launchers, links to web pages, and shortcuts to directories. It also provides an easy way to organize and tile the icons that are already visible on the desktop.

**Adding new items to the desktop:** All you need to create new links and launchers is located in the context menu that appears when you right-click on the desktop. To add a link to a directory, select New Launcher. In the dialog box that appears, type the name of the directory, select directory as the type, and click OK. An icon will appear on the desktop that, when opened, will launch the GNOME File Manager to display the contents of the directory you specified. The same process applies to creating a URL link. Simply choose New Launcher from the context menu and select Link as the type in the window that appears. Then, enter the address of the web page you wish to access in the URL field. Once you click on OK, an icon will appear on the desktop that will launch netscape and navigate to the page that you chose.
To create shortcuts to your favorite programs, called launchers, choose New Launcher from the context menu. A window will appear. Enter the name of the launcher, a comment that will appear when you place your mouse pointer over the icon, and the path to the location of the program in the filesystem. Then choose the type of launcher. Typically here you should choose application. If the program that you specified must be run inside a terminal window, click on the check box labeled “Run in Terminal”. Finally, choose an icon for your launcher by clicking on the box that by default displays “No Icon”. A selection of icons will appear. To pick one, simply double click on it. To create the icon, click OK. To cancel the operation, click on Close.

Removing desktop items: To remove items from the desktop, simply drag them to the trash can. You can also right-click on them and choose delete. A dialog box will appear to confirm your decision to delete. Click OK to delete the item.

Panel Context Menus

Adding items to the panel: The panel can also host launchers. To add a launcher to the panel, right-click anywhere on the panel. In the context menu that appears choose Panel -> Add to panel. In the sub-menu that opens you will see a list of objects that can be added to the panel. Select launcher to create a shortcut icon. This will open the Create Launcher applet described above that will create the launcher for you and place it in the panel.

The panel can also host a variety of other objects including menus, drawers, applets, and special control buttons. To add one of these objects simply select them from the Panel -> Add to panel submenu that appears when you right-click on the panel. Once you choose one from the sub-menu, it will appear in the panel.

Moving items in the panel: You can adjust the position of any object in the panel simply by right-clicking on the object and choosing move from the context menu that appears. Your mouse pointer will change to a set of crossed arrows and the movements of the mouse will shift the position of the object that you selected within the panel. Click the left mouse button when you wish to place the object.

Removing panel items: To remove an object from the panel, right click on it and choose remove from panel. The object will disappear from the panel without confirmation.
Customizing your K desktop environment

Using the K Control Center

Many of the options that control the look-and-feel of the K Desktop Environment are configurable through the K Control Center. To start this configuration tool, choose “Control Center” from the Application Starter. In its default configuration, the K Control Center presents a two-paned window displaying a hierarchy of configuration options on the left and a display panel on the right. To start configuring a particular aspect of your machine, click on the item in the hierarchy on the left and a configuration page will open on the right. The various categories of configurable items are:

- **Appearance and Themes**: changes the appearance of your desktop environment.
- **Desktop**: customizes look and feel of the desktop.
- **Information**: provides information about your system.
- **Internet & Network**: changes a number of network preferences.
- **KDE Components**: configures the behavior of KDE tools.
- **Peripherals**: allows you to alter the behavior of the mouse and keyboard.
- **Personalization**: changes options dealing with localization, security, and accessibility for the physically challenged.
- **Power Control**: used to alter battery and energy conservation options.
- **Regional & Accessibility**: configures localization and access tools for the physically challenged.
- **Security & Privacy**: changes SSL and password properties.
- **Sound & Multimedia**: configures multimedia options.
- **System Administration**: sets path settings and help behavior.
- **Web Browsing**: provides access to Internet options.

To configure an item, click on it in the left pane and make the desired adjustments in the form that appears on the right. To save the changes that you have made, click on the Apply button at the bottom of the form. The Reset button undoes the changes made with the Apply button. Clicking on Use Defaults will restore the system’s default settings.
Customizing the Desktop

Due to its drag-and-drop capabilities, the KDE desktop is highly configurable and easy to use. You can organize your workspace by adding the items that you use most often and removing those that you do not need.

Adding Items to the Desktop: Adding and removing icons to and from the desktop is primarily performed through drag-and-drop. To add an icon to the desktop, you can drag it from another location, such as the Konqueror browser or the Kicker. After dropping the icon, you will be presented with a dialog box offering three choices: copy, move, and link. For most documents and probably all applications you will want to choose link to create a symbolic link to the item in its original location. This saves disk space when compared to copying items and will assure that you are always working from the same version of a document regardless of how you opened it.

You can also add items to the desktop by placing them in your desktop folder, which is located in ~/Desktop, where ~ is your home directory. You can save files, applications, and symbolic links in this folder and they will appear on the desktop.

Adding Internet URLs to the Desktop: You can add shortcuts to your favorite internet sites on the desktop by right-clicking on an empty space in the desktop and choosing Create New->Link to Location (URL) from the context menu. Enter the URL in the dialog box that appears and the icon will appear on the desktop.

Removing Items from the Desktop: You can remove desktop items either by dragging the icons to the trash icon, or by removing the items from ~/Desktop.

Using The Autostart Folder

KDE provides an easy way to start programs when a KDE session starts through the autostart folder located by default on the desktop. All you have to do to get programs to start automatically is to drag them into the autostart folder. When asked, choose Link to create a symbolic link to the desired application in the folder rather than copying the application. This will save a great deal of disk space if you use this feature extensively.
Customizing the Kicker

The Kicker is highly customizable and will accept a number of shortcuts and applets. Access to the configuration of the Kicker is found in the Application Starter under the Panel submenu.

Adding an Item to the Kicker: To add items to the kicker, simply click on the K Menu and choose Add. This will expose a number of submenus containing different types of items that can be added to the Kicker. For example, to add a shortcut to a program, choose Add->Application. You will see the top level of the application starter again. Go through the menus to find the entry for which you want to create the shortcut, such as Home directory or Konqueror. Click on the program you want and a shortcut for it will appear in the Kicker.

Moving Items in the Kicker: To move an icon to another area in the Kicker, right-click on it and choose Move or simply click on it with the middle mouse button. The icon will now follow your mouse. Move the pointer to the location in the Kicker where you would like to place the icon and click on the left button. The icon will now be in its new position.

Removing Items: You can remove items from the Kicker by right-clicking on them and choosing Remove.

Customizing the K Menu

You can easily add and remove programs and shortcuts from the K Menu using the K Menu Editor. To start it, use the application starter and choose System->Menu Editor. A window will open showing the existing K Menu on the left, and an Add Menu Entry form on the right.

Adding an Item: To add an item, click on the submenu in which you would like to add the new application in the left panel. Once it is highlighted, click on the New Item icon in the toolbar, or optionally, choose File->New Item in the menu bar. A dialog box will appear. Enter the name of the shortcut that you wish to create and click OK. The new shortcut will appear in the hierarchy on the left. Click on it. The form to the right will display the name of the shortcut that you just created. In the Command blank on this form, fill in the command that starts this application from the command line. If you wish to change the icon from the default ‘unknown’, click on the icon to the right of the dialog to expose a variety of icon choices. Finally, if your program must be run in a terminal window, click on the Run in a Terminal checkbox. Click on Apply to save your work and create the shortcut.
Adding a Submenu: To create a new submenu, click on the position in the hierarchy on the left where you would like to place the new submenu. Click on New Submenu in the toolbar. In the dialog that appears, enter the name of the submenu you wish to add. Click OK and it will appear in the K Menu.

Deleting an Item or a Submenu: To delete an item or submenu, highlight it in the hierarchy on the left and click the delete button in the toolbar. It will be removed from the K Menu.

Organizing menu items: To move items and submenus simply drag and drop them to new directories in the hierarchy on the left. They will be automatically repositioned in the K Menu.

Customizing your X environment

To understand where to go and what to do to customize your X environment, you should understand how your environment is created. When you log in on your workstation, the system reads your .cshrc and .login files. The latter normally executes a file called /usr/local/lib/profiles/login.def. This file sets paths and environment variables that enable your personal .login to start X without any problems.

Your default .login file can set the XVERSION environment variable and invokes the program xinit, which starts X. xinit reads a file called .xinitrc, which starts the window manager, runs xrdb (a program that loads user preferences specified in the file .Xdefaults into the resource database), a clock, an xterm, and any other applications you specify. The Motif window manager (mwm) reads the file .mwmrc as it starts.

You can change many aspects of the default environment. Some of the more common adaptations are:

- Window size and placement
- Text font
- Background image
- Applications automatically started when X starts, for example, a telnet window or a clock program.
- Other window attributes, such as autoraise, reverse video, window title, and border width.
Mechanisms for changing the default environment

You can change your X environment by:

- **Using command line switches.** Use switches when starting an application from the command line, or put them in your .xinitrc file, which contains commands executed during start-up. See "Making changes at the command line" on page 32 for more on using command line switches.
- **Changing the .xinitrc file.** xinit reads and executes commands in .xinitrc when it starts X. Typically, the window manager, an xterm window, and a clock are started in this file. However, you can start many other clients from this file. For more on adding applications to .xinitrc, see page 35.
- **Storing application resources via xrdb.** xrdb is a program that allows you to store user preferences (called resources) for X programs in a central, server-maintained database. Using resources eliminates the need to specify command line switches every time you start an application. For more on specifying resources, see page 37.
- **Modifying mwm.** The .mwmrc file can be used to modify menu items and how or where menus are raised. mwm resources can be used to modify other behavior, such as “autoraising” and what window decorations are present for specific clients. For more on modifying mwm, see the section starting on page 40.

Customization examples

This section illustrates a few of the most frequent environment modifications. Program manual pages contain information on program-specific customizations.

Making changes at the command line

You can use command line switches to modify all X applications. This mechanism affects only the current invocation of the application. For example, if you want to bring up a clock from the command line, you could type xclock & and the standard clock would appear. You could modify the clock’s normal appearance and function by specifying some switches; the following line places a reverse-video digital clock five pixels from the lower right corner:

```
xclock -geometry -5-5 -digital -rv -chime &
```

The clock will chime once on the half hour and twice on the hour.
Find application switches in the manual page for each application. You may notice that many manual pages say their applications “accept all of the standard X Toolkit command line options”. You’ll find the complete list of these options in the man page `X(1)`, which you can view by entering `man X` at the command prompt. See the following table for some of the more commonly used options.

### Common command line options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-geometry geometry</code></td>
<td>Specify the initial size and location of the window. See the next section for more on the geometry switch.</td>
</tr>
<tr>
<td><code>-fn, -font font</code></td>
<td>Specify the font to use for displaying text.</td>
</tr>
<tr>
<td><code>-iconic</code></td>
<td>Specify that the application’s windows should initially appear as icons only.</td>
</tr>
<tr>
<td><code>-rv, -reverse</code></td>
<td>Specify that the program should simulate reverse video, usually black-on-white. <code>-rv</code> means it should not.</td>
</tr>
<tr>
<td><code>-name string</code></td>
<td>Specify the name under which to find resources for the application. This switch also changes the title displayed in the title bar. See page 38 for using “name” to customize windows.</td>
</tr>
<tr>
<td><code>-title string</code></td>
<td>Specify the title to be used for the window. Note that this switch does not change the instance name of the window. See page 38 for more on instances.</td>
</tr>
</tbody>
</table>

### Changing window size and/or placement

You can change the size or placement of your login or other windows with the geometry resource:

```
application -geometry WxH+X+Y
```

`W` is the width of the window in pixels or characters, depending on what the application expects; `H` is the height. `+X` and `+Y` are offset values in pixels from the upper left corner of the screen.

For example, specify an `xterm` window’s placement in this way:
xterm -geometry 80x45+1-1

This window is 80 characters wide, 45 characters high and is offset 1 pixel from the lower left corner.

Specify any of the four corners of the screen as the offset origin by using positive or negative values for X and Y. Specifically,

- \(+X+Y\) is offset from the upper left corner of the screen
- \(-X+Y\) is from the upper right corner
- \(+X-Y\) is from the lower left corner
- \(-X-Y\) is from the lower right corner

We recommend offset values of at least one pixel to keep windows from the edge of the display screen.

Both size and placement specifications are optional: that is, you can simply specify \(X\) and \(Y\), e.g. “-geometry +5+5”, or just a width and height, e.g. “-geometry 80x10”.

You can use the geometry switch for most \(X\) applications. Specify an \texttt{xclock} window’s size and placement, for example, with the line:

\texttt{xclock -geometry 100x80+20+1}

Note that \texttt{xclock} interprets the width and height numbers in pixels. To specify a default geometry for all instances of an application, you can use the \textit{geometry} resource in your \texttt{.Xdefaults} file. See page 37 for more on specifying resources.

**Background image for the root window**

To add an image to your root window, use \texttt{xsetroot -bitmap pathname}

One source of images is this directory:

\texttt{/usr/local/include/X11/bitmaps}

For example, try the following command:

\texttt{xsetroot -bitmap /usr/local/include/X11/bitmaps/escherknot}

Try the file “black” in the same directory. To display a particular image on your root window every time you start \(X\), put the \texttt{xsetroot} command in your \texttt{xinitrc} file. See the \texttt{xsetroot} manual entry for more information on that command.
Starting applications from your .xinitrc file

*xinit* reads the file .xinitrc in your home directory when it starts *X*. .xinitrc contains applications that you wish to start automatically. The default .xinitrc contains lines that start a window manager, a clock, and an *xterm* window. Add to this file any other application that you would like to start at login. You can also change an application’s default characteristics. For example, if you want to change the size of an *xterm* window, modify the -geometry switch to the line that invokes *xterm*. Or, if you use the *X* calculator often, add the following line to invoke it each time you log in:

```
xcalc -geometry -5-5 &
```

**Caution:** Be sure to put your personal additions *before* the command that starts your login *xterm*. If you don’t, *xinit* will not reach them and they will never be started. Also, be sure to end each command line with an ampersand (&) so that the command is run in the background. If you don’t do this, *X* will wait for the command to finish and won’t proceed to the next line in the .xinitrc file.

Running *X* clients from foreign hosts

*xhost* allows you to run an *X* program on a foreign host but display the window on your workstation’s screen. You could do this, for example, if the foreign host runs software that yours does not. To run the software, you add the desired host to your workstation’s access list:

```
xhost hostname
```

You then open an *xterm* window, *telnet* to the host, and run the desired software. Because you used *xhost*, the software will be able to open new windows on your workstation.

*On being reasonable:* It is considered both inconsiderate and unreasonable to run *X* programs on general-purpose machines. The primary function of most general-purpose machines is to serve as a mail drop for a large number of users. If you use such a machine to run *X* programs, you interfere with the productivity of a large number of users by slowing down their access to mail. If you have any questions on the reasonable use of *xhost*, please send email to the Help Center (help@cs.cmu.edu).

**Caution:** When you add a host to the access list it gives everyone on that host access to your workstation display. That is, everyone on that host can potentially see all of your windows and possibly all of your keystrokes. *Never* type “xhost +”. If you do, you make your machine accessible to all hosts.
To remove a host from the access list, type

```
xhost -host
```

Generally, you should provide `xhost` access only as long as it takes to start up the program on the remote machine. That is, you’ll perform the following steps:

- Run `xhost host` on your workstation
- Start the X program you want to run on the remote machine
- Run `xhost -host` as soon as the remote program’s window is displayed on your screen
4 Advanced customization

You can customize X clients in many ways. Once you are familiar with X basics, you can move on to advanced customizations. These include:

- Specifying resources with class and instance names
- Modifying the window manager
  - Changing mwm resources
  - Assigning mouse/key bindings to window manager functions
  - Generating personal menus
- Specifying/arranging window-specific fonts

Specifying resources

You may want to change the appearance or behavior of an X application. For example, you may want the xcalc calculator to use reverse Polish notation, or you may want xterm windows to use reverse video (usually white letters on a black background). To learn about your preferences, X examines your resource database. Entries in this database are known as resources.

You can change the contents of the resource database by using the xrdb program. In your .xinitrc file, you’ll find the following line:

```
xrdb .Xdefaults
```

Each time you log in, xrdb reads the .Xdefaults file and stores the specified resources in the resource database, which all X clients check when they start. The file .Xdefaults is normally in your home directory and contains standard SCS resource specifications copied from /usr/local/lib/profiles/Xdefaults. You can specify your own resources by editing the .Xdefaults file. Remember that you can copy the original files from their sources if something goes wrong.

Each resource specifier usually contains three components: the application name, the option you want to modify, and the value for the resource.

```
application_name*resource_option=value
```

For example, the following line in .Xdefaults activates the scroll bar in an xterm window:
Before specifying a resource for a program, you must know what resource options it supports. The man page for the program will give you that information. For example, while most programs support the resource option “geometry,” which allows you to set the window size, few would likely support the option “rpn,” which instructs xcalc to accept input in reverse Polish notation.

Note: The term “resource” is used ambiguously in X manual entries to refer to one of two things:

- The name of the behavior or object you wish to modify—what we have called a “resource option”
- The complete specification of your preference, including the application name, the resource option, and the value for the option.

Specifying class names of applications

Use class names to make a resource apply every time an application is invoked. An application’s class name is formed by capitalizing the first letter of its name. For applications starting with “X,” capitalize the “X” and the first letter of the next word, e.g. “XTerm” and “XClock.”

Suppose you want reverse Polish notation to apply every time you invoke xcalc. In the .Xdefaults file, specify the class name “XCalc” as follows:

```
XCalc*rpn:on
```

Type `xrdb .Xdefaults` to add the changes to the database. xcalc will now use reverse Polish notation.

Using instance names of applications

While you use a class name to describe how you usually want an application to function, you can use an instance name to establish exceptions to the class name specification. Instance names allow you, for example, to distinguish two otherwise identical xterms and request different resources for each. Instance names are by default the same as the application name and are written in all lower case letters. When you invoke a program, you use the “-name” switch to specify an instance name, thus creating a special version of that program. For example, the xterm window created from the line:

```
XTerm*scrollBar:on
```
xterm &

has the default instance name “xterm” and the class name “XTerm.” But an xterm window created with the following line has the instance name “myxterm” and the class name “XTerm”:

```
xterm -name myxterm &
```

The latter window would be affected by resources you have specified in Xdefaults with the instance name “myxterm,” for example:

```
myxterm*geometry:80x54
```

Consider the following example. Suppose you want to have three xterm windows on your screen, named “xterm” (the default), “myxterm,” and “telnet,” with the following attributes:

- All are in reverse video.
- Those instances with the default application name “xterm” should use the font “rk14”
- Those named “myxterm” should use the font “9x15”.
- Those named “telnet” should be in a certain size and location.

To satisfy the above requirements, you could add the following resources to the .Xdefaults file:

```
XTerm*reverseVideo:on
xterm*font:rk14
myxterm*font:9x15
telnet*geometry:80x48-1+1
```

Remember to run `xrdb` again (see page 37). Then bring up the windows:

```
xterm &
xterm -name myxterm &
xterm -name telnet &
```

The first resource, XTerm*reverseVideo:on, will affect all windows brought up by xterm. The remaining resources, specified with instance names, will affect only windows with the corresponding names.

### Applying resources to all windows

If you want all windows, regardless of their applications, to have a certain attribute, you can specify it in the following fashion:

```
*resource_option:value
```

For example, to make all applications use the font rk14, specify:

```
*font:rk14
```
Class and instance names of resource options

If you have a color monitor, you will need to know about class and instance names for resource options as well as for applications. As with applications, resource option class names begin with a capital letter, instance names with a lower case letter. For many options, there is no distinction between instance and class. The following lines have identical functionality:

    xterm*scrollBar:on
    xterm*ScrollBar:on

However, when used on a color monitor, many text applications have some notion of background, foreground, border, pointer, and cursor color. For example, on a monochrome monitor, it makes sense to define only the classes:

    application*Background:white
    application*Foreground:black

On a color display, however, one might distinguish the border, pointer, and cursor from other elements of the foreground:

    application*Background:white
    application*foreground:red
    application*borderColor:blue
    application*pointerColor:green
    application*cursorColor:black

See Using and Specifying X Resources [1] for further information, available online in

    /afs/cs/archive/X/V11R4-source/mit/doc/tutorials/resources.txt

Modifying the window manager, mwm

Using resources

You can modify many aspects of the window manager by specifying its resources in .Xdefaults. A complete list of mwm resource options can be found in the man page mwm(1). Here we list a few examples.

keyboardFocusPolicy

This resource option can be set to “explicit” (the default) or to “pointer.” Set to “pointer,” keyboardFocusPolicy causes input focus to automatically go to whatever window the mouse cursor is in:
By itself, this line does not tell X to bring the window to the front: it only says to make the window active. You will find the `keyboardFocusPolicy` option defined but commented in your .Xdefaults file. Remove the “!!” from the beginning of the line to get pointer focus.

**focusAutoRaise**

If `keyboardFocusPolicy` is set to “pointer,” you should also set `focusAutoRaise` to “True” to raise the active window to the top of the stack:

```
Mwm*focusAutoRaise:True
```

**clientDecoration**

This resource option helps you control window “decoration”. You can tell `mwm` what components to place around your windows using one or more of the following values for `clientDecoration`:

- all —Include all decorations
- border—Window border
- maximize—Maximize button (includes titlebar)
- minimize—Minimize button (includes titlebar)
- none—No decorations
- resizeh—Border resize handles (includes border)
- menu—Window menu button (includes title bar)
- title—Title bar (includes border)

Some values automatically include others when they are selected. For example, “Maximize” automatically includes a title bar with the maximize button. Specify the resource as follows:

```
Mwm*Client.clientDecoration:list
```

where `Client` is some X program and `list` is some combination of the values given above. The sign of the items in the list is important. A plus or no sign indicates that `mwm` should only use the listed decorations. A minus sign indicates that `mwm` should start with the default (all) and subtract only the items in the list. For example:

```
Mwm*XClock.clientDecoration: border resizeh
Mwm*XTerm.clientDecoration: -maximize
```

The first line tells `mwm` that all xclock windows should have only the window border and resize handles. The second line tells `mwm` that xterm decoration should exclude only the maximize button.
Note that the client names are specified as class names (using upper case) in the examples. They could be specified as instance names to further customize mwm. See the previous section on X resources for more on class and instance names.

**iconDecoration**
The values for this option are label (only the icon label is displayed), image (only the image part is displayed), or both. An additional value is activelabel, which provides a non-truncated label when the icon is selected. To save screen space, the SCS default is “label”:

```
Mwm*iconDecoration: label
```

**fontList**
This resource option specifies the font used in the window decoration. For example:

```
Mwm*fontList: -adobe-times-bold-r-normal-*-*-*-*-*-*-*-*
```

See page 45 for directions on seeing available fonts.

**windowMenu**
This resource option allows you to use application-specific window menus. For example, you may want different functionality for your xcalc window and define a different menu for it (see page 44 for how to define a menu). To get mwm to use your new menu for xcalc windows, you must specify the windowMenu option; for example:

```
Mwm*XCalc*windowMenu: XcalcMenu
```

**Using the resource description file: .mwmrc**
You can modify or create menus and change mouse/keyboard buttons that invoke functions by modifying the mwm resource description file .mwmrc (in your home directory). This file contains resource descriptions to be used by mwm that cannot be easily encoded in .Xdefaults. Your .mwmrc file is organized in three sections: menu definitions, key bindings, and button bindings. We provide here only a brief overview of the .mwmrc file. Type man mwm or see the OSF/Motif Programmer’s Guide [2] or the X Window System User’s Guide: OSF/Motif Edition [3] for a complete description of mwm menus and button/key bindings.

**Defining window menus**
The first section of .mwmrc contains definitions for the menus that appear when you click the left button in the root window or title bar of a regular window. Root menu definitions are in the following form:
**Chapter 4 Advanced customization**

*label* function

For example, the following line means that when you select “Circle Up” on the root menu, the window on the bottom will come to the top:

"Circle Up" f.circle_up

The window menu description defines alternate ways of carrying out functions; these definitions are in the following form:

*label* mnemonic accelerator function

For example, the following line means that you can make a window move by selecting “Move window,” by typing “M,” or by pressing [ALT], then F3:

"Move window" _M Alt<Key>F3 f.move

You can delete or add to these menus following the syntax above. You’ll find a complete list of mwm functions in the man page; for a partial listing, see below.

### Some mwm functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f.circledown</td>
<td>Drop the top window to the bottom of the window stack. Useful when a window is obscuring the view of other windows.</td>
</tr>
<tr>
<td>f.exec or ! shell command</td>
<td>Execute the following shell command. Use ! in place of “f.exec.”</td>
</tr>
<tr>
<td>f.refresh</td>
<td>Redraw all windows.</td>
</tr>
<tr>
<td>f.refresh_win</td>
<td>Redraw a client window.</td>
</tr>
<tr>
<td>f.kill</td>
<td>Eliminate window.</td>
</tr>
<tr>
<td>f.maximize</td>
<td>Enlarge window to maximum size.</td>
</tr>
<tr>
<td>f.minimize</td>
<td>Iconify window.</td>
</tr>
<tr>
<td>f.menu string</td>
<td>Assign a pull-down menu named string to a pointer button.</td>
</tr>
<tr>
<td>f.restart</td>
<td>Exit and restart mwm to implement changes to .mwmrc.</td>
</tr>
<tr>
<td>f.raise_lower</td>
<td>Raise window if partially obscured; otherwise lower to the bottom.</td>
</tr>
</tbody>
</table>

**Binding keys to functions**

Keyboard events can also be bound to mwm functions. Key binding definitions are in the following form:

*key* context function

*context* indicates where the pointer must be for the key binding to be effective. *context* must be one of:
For example, the following states that pressing [META] (or [ALT] or the corresponding key on your keyboard) then [SPACE] brings up a menu in a window or icon:

\[
\text{Meta<Key>space} \quad \text{icon|window} \quad \text{f.post_wmenu}
\]

Binding buttons to functions
The button bindings section defines what functions should be executed when you use mouse buttons while the cursor is in a specific area on the screen. Functions can be bound to any mouse button combination. Button binding definitions are in the following format:

\[
[\text{modifier}] \text{button} \quad \text{context}[|\text{context}] \quad \text{function} \quad [\text{argument}]
\]

\text{modifier} is optional and can be one of:

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl</td>
<td>Control Key</td>
</tr>
<tr>
<td>Shift</td>
<td>Shift Key</td>
</tr>
<tr>
<td>Meta</td>
<td>Meta /Alt /ComposeCharacter Key</td>
</tr>
<tr>
<td>Lock</td>
<td>ShiftLock Key</td>
</tr>
</tbody>
</table>

\text{Button} is specified in a form like “Btn1Down,” meaning “press the first button.” You’ll find a complete listing of button events in the manual pages for \text{mwm}.

Creating a menu
To create a menu, first define the new menu. The syntax is:

\[
\text{Menu} \quad \text{menu}_\text{name} \\
\{} \\
\quad \text{“label”} \quad \text{function} \\
\quad \text{“label”} \quad \text{function} \\
\quad \ldots \\
\quad \text{“label”} \quad \text{function} \\
\}
\]

Replace \text{menu}_\text{name} with an appropriate menu name. Each value for \text{label} specifies a name that will appear on the menu listing, while \text{function} specifies the \text{mwm} function to be executed when the
corresponding item is selected. Shell commands can be executed using the “f.exec” function and cannot include a carriage return. “!” can be used as a substitute for “f.exec”. So your menu may look like this:

```plaintext
Menu MyMenu
{
“My Menu” f.title
no-label f.separator
“Refresh” f.refresh
“V.GP” ! “xterm -geometry 100x56+210+1 -e telnet v.gp.cs.cmu.edu &”
“I.GP” f.exec “xterm -e telnet i.gp.cs.cmu.edu &”
}
```

“f.title” posts the menu title; “f.separator” puts a menu separator between the title and the menu choices. Choosing the menu item “Refresh” will redraw the screen. Choosing “V.GP” or “I.GP” will execute the shell command line, starting an xterm that is brought up running a telnet program.

Next, assign the menu to a mouse button/modifier key combination, using the f.menu function. Put the new line in the section called “Button bindings” in .mwmrc, under the default bindings. For example:

```plaintext
<Btn3Down> root f.menu MyMenu
```

tells mwm that when you press the right mouse button in the root window, it should execute the function “f.menu” and the menu it should bring up is “MyMenu”. Or for a “cascading” menu, invoke it from a window or root menu. Add the following line to the appropriate menu:

```plaintext
“My menu” f.menu MyMenu
```

See the manual page mwm(1) for more information.

### Changing window text font

#### Listing available fonts

The program xlsfonts lists the available fonts:

```plaintext
xlsfonts | more
```

If you know what type of fonts you are interested in, use the -fn switch and wildcard characters to see that type of font. For example, to see all Courier fonts, type:

```plaintext
xlsfonts -fn “*courier*”
```

Be sure to enclose the argument to -fn in double quotes.
The default font for displaying text is 8x13, an alias for the font whose full name is

-misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1

Other aliases are defined for commonly-used font specifiers:

- 5x8  -misc-fixed-medium-r-normal--8-80-75-75-c-50-iso8859-1
- 6x10 -misc-fixed-medium-r-normal--10-100-75-75-c-60-iso8859-1
- 7x13 -misc-fixed-medium-r-normal--13-120-75-75-c-70-iso8859-1
- 7x13bold-misc-fixed-bold-r-normal--13-120-75-75-c-70-iso8859-1
- 8x13 -misc-fixed-medium-r-normal--13-120-75-75-c-80-iso8859-1
- 9x15 -misc-fixed-medium-r-normal--15-140-75-75-c-90-iso8859-1
- k14  -*fixed-medium-r-normal--14-*-*-*-*-jisx0208.1983-*
- rk14 -*fixed-medium-r-normal--14-*-*-*-*-jisx0201.1976-*
- r16  -*fixed-medium-r-normal--16-*-*-*-*-jisx0201.1976-*
- rk16 -*fixed-medium-r-normal--16-*-*-*-*-jisx0201.1976-*
- kana14-*fixed-medium-r-normal--14-*-*-*-*-jisx0201.1976-*

The complete list of font aliases is in the file /usr/local/lib/X11/fonts/misc/fonts.alias

**Viewing fonts**

There are two ways to view fonts: xfd and xfontsel. xfd will accept font aliases such as “8x13”, “9x15”, or “rk14”. For example, to see the font “8x16romankana”, you can type:

```
xfd -fn 8x16romankana
```

See the xfd(1) man page for more information.

xfontsel will not accept aliases such as “8x13” but will accept wild card characters. For example, if you want to see the fonts that have the string “b&h” in them, type:

```
xfontsel -pattern "b&h"
```

See the xfontsel(1) man page for more information.

**Specifying fonts**

When you find a font that you like, use the “font” resource option to change fonts for different applications:

```
xtterm -fn rk14 &
kanaterm*font:rk14
```

**Adding font paths**

The default font directory paths are:
If you want to add additional font directories to your font path list, use the program `xset`. See the man pages for this command and the X Window System User’s Guide: OSF/Motif Edition [3] for more information. Be careful to use only fixed-width fonts (those with either “m” or “c” near the end of the string describing them) for `xterm` windows, or else backspace/delete may not work quite right, nor will tabular information be displayed properly.
# Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active window</strong></td>
<td>Window currently accepting keyboard and mouse input. There can be only one active window per screen. Make a window active by placing the mouse pointer in it and clicking the left mouse button.</td>
</tr>
<tr>
<td><strong>Click</strong></td>
<td>Press and release a mouse button.</td>
</tr>
<tr>
<td><strong>Client</strong></td>
<td>Any application that relies on the (X) server for IO.</td>
</tr>
<tr>
<td><strong>Cut and paste</strong></td>
<td>Select text region and copy it using the mouse.</td>
</tr>
<tr>
<td><strong>Double click</strong></td>
<td>Quickly press and release a mouse button twice.</td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>Your workstation’s video screen. A workstation may have more than one display. Displays are numbered, starting with 0.</td>
</tr>
<tr>
<td><strong>Geometry</strong></td>
<td>Width, height, and position of application window, measured in pixels or characters, depending on what the application expects.</td>
</tr>
<tr>
<td><strong>Iconify a window</strong></td>
<td>Reduce a window to a small image called an icon, or to an entry in the icon manager window.</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>Information given to a program via keyboard or mouse.</td>
</tr>
<tr>
<td><strong>Listener</strong></td>
<td>The active window.</td>
</tr>
<tr>
<td><strong>Mouse cursor</strong></td>
<td>The symbol in an xterm window that looks like a large capital “I”. Its position is controlled using the mouse, and its appearance differs according to the application.</td>
</tr>
<tr>
<td><strong>Press</strong></td>
<td>Press down and hold a key or a mouse button.</td>
</tr>
<tr>
<td><strong>Release</strong></td>
<td>Release a key or a mouse button.</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td>Specifications of window appearance or program behavior. The man pages list resources for each (X) program.</td>
</tr>
<tr>
<td><strong>Resource database</strong></td>
<td>Storage place for user preferences. It is accessible to all clients of the (X) server, regardless of the host machine.</td>
</tr>
<tr>
<td><strong>Root window</strong></td>
<td>Large, normally grey window that appears to be under the other windows and that almost covers your screen as X starts.</td>
</tr>
<tr>
<td><strong>Scrolling</strong></td>
<td>The process by which text in xterm windows may be moved up and down using scrollbars on the left side of the window.</td>
</tr>
<tr>
<td><strong>Terminal emulator</strong></td>
<td>Program such as xterm that behaves like a terminal: accepts text input/output, mouse clicks, etc.</td>
</tr>
<tr>
<td><strong>Titlebar</strong></td>
<td>The horizontal bar at the top of a window.</td>
</tr>
<tr>
<td><strong>Window</strong></td>
<td>Visually, a subdivision of your screen. A window may act as a terminal, or display output from a program, such as the window containing the clock. When the mouse cursor is in a window, that window is the “active” window and receives all keyboard and mouse input; its titlebar is a horizontal grey rectangle.</td>
</tr>
</tbody>
</table>
X utilities

Short descriptions of many X utilities are listed below. See the corresponding man pages for more details.

**Miscellaneous Utilities**

- **xbiff**: Displays a little image of a mailbox: the flag on the mailbox goes up and the mailbox beeps when new mail arrives.
- **xcalc**: Emulates a TI-30 or an HP-10C calculator.
- **xclock**: Displays the time in analog or digital form. You may specify how often the time is updated. (See `/usr/local/lib/profiles/xinitrc` for samples.)
- **xhost**: Adds or deletes hosts in the list of machines that are allowed to make connections to your X server.
- **xinit**: Starts the X Window System server: normally called from your `.login` file. (See `/usr/local/lib/profiles/login`, the standard login file.)
- **xpostit**: Displays on-screen Post-it® notes. All six sizes of Post-it notes may be displayed, edited, and saved to disk files. Notes may be resized.
- **xrefresh**: Repaints all or part of your screen: useful when system messages have cluttered your screen.
- **xstdcmap**: Selectively defines standard colormap properties. It is intended to be run from a user’s X startup script to create standard colormap definitions in order to facilitate sharing of scarce colormap resources among clients.
- **xterm**: Provides DEC VT102 and Tektronix 4014 compatible terminals for programs that can’t use the window system directly.
- **xloadimage, xsetbg, xview**: Displays images in an X11 window or loads them onto the root window. See the manual entry for supported image types.
A variety of image manipulations can be specified, including gamma correction, brightening, clipping, dithering, depth-reduction, rotation, and zooming.

**xv** Displays images in the GIF, PBM, PGM, PPM, X11 bitmap, JPEG, Sun Rasterfile, and PM formats on 1-, 4-, 6-, 8-, 16-, 24-, and 32-bit X displays. **xv** will also read compressed versions of these files.

## Adjusting your X environment

**xset** Seta various user preference options of the display such as: bell volume, key click, font path, mouse tracking, autorepeat, and screen saver parameters.

**xsetroot** Tailors the appearance of the background (“root”) window on a workstation display running X.

**xmodmap** Configures the keyboard to your personal tastes; usually run from your “.xinitrc” file.

**xrdb** Stores preferences about color, fonts, etc. in your X resource database.

**xscreensaver** Locks the local X display (and server) until a password is entered.

**xmag** Magnifies portions of the screen.

**xkill** Kills a client by its X resource: forces the X server to close connections to clients. (Dangerous, but useful for aborting programs.)

## Cut buffer and clipboard

**xclipboard** Collects and displays text selections that are sent to the clipboard by other clients. It is typically used to save clipboard selections for later use. It stores each clipboard selection as a separate string, each of which can be selected.

**xcutsel** Copies the current selection into a cut buffer and makes a selection that contains the current contents of the cut buffer. It acts as a bridge between applications that don’t support selections and those that do.
**X Window “dumps”**

- **xwd** Stores a window images in a specially formatted dump file. This file can then be read by various other X utilities for redisplay, printing, editing, formatting, archiving, image processing, etc.

- **xwud** Displays (in a window) an image saved in a specially formatted dump file, such as produced by `xwd(1)`.

- **xpr** Formats a window dump file (as produced by `xwd(1)`) for output on PostScript printers, the Digital LN03 or LA100, the IBM PP3812 page printer, the HP LaserJet (or other PCL printers), or the HP PaintJet.

- **xdpr** Uses the commands `xwd(1)`, `xpr(1)`, and `lpr(1)` to dump an X window, process it for a particular printer type, and print it out on the printer of your choice. This is the easiest way to get a printout of a window.

**X Status Utilities**

- **appres** Lists the resources seen by an application of the specified class and instance name. Used to determine which resources a particular program would load. For example,
  
  ```
  % appres XTerm
  ```

  would list the resources that any xterm program would load. To also match particular instance names,
  
  ```
  % appres XTerm myxterm
  ```

  If no application class is specified, the class -NoSuchClass- (which should have no defaults) is used.

- **xauth** Displays and edits the authorization information used in connecting to the X server. This program is usually to extract authorization records from one machine and merge them in on another (as is the case when using remote logins or to grant access to other users).

- **xdpyinfo** Displays the capabilities of a server, the predefined values for various parameters used in communicating between clients and the server, and the different types of screens and visuals that are available.

- **xev** Creates a window and then asks the X server to send it notices called events whenever anything happens to the window (such as being moved, resized, typed in, clicked in, etc.). It is useful for seeing what causes events to occur and to display the information that they contain.
**X utilities**

**xlsatoms** Lists the interned atoms. By default, all atoms starting from 1 (the lowest atom value defined by the protocol) are listed until unknown atom is found. If an explicit range is given, xlsatoms will try all atoms in the range, regardless of whether or not any are undefined.

**xlsclients** Lists the client applications running on a particular display.

**xprop** Displays window and font properties in an X server.

**xshowcmap** Displays the contents of the currently active colormap in a window. Shows a square for every color currently defined in the server’s active colormap. The number of squares is the number of colormap cells the server supports.

**xwininfo** Displays information about windows.

**X Resource utilities**

**editres** Displays the full widget hierarchy of any X Toolkit client that speaks the Editres protocol. In addition editres will help the user construct resource specifications, allow the user to apply the resource to the application and view the results dynamically.

**listres** Generates a list of a widget’s resource database. The class in which each resource is first defined, the instance and class name, and the type of each resource is listed.

**Font Utilities and Conversion programs**

**xfd** Creates a window containing the name of the font being displayed, a row of command buttons, several lines of text for displaying character metrics, and a grid containing one glyph per cell. The characters are shown in increasing order from left to right, top to bottom.

**xfontsel** Allows you to examine samples of fonts and determine the full name (X Logical Font Description) for a font, using a “point and click” interface.

**xlsfonts** Lists names of fonts available.
Bitmap utilities

**bitmap, bmtoa, atobm**

*bitmap* creates or edits rectangular images made up of 1's and 0's. Bitmaps are used in X for defining clipping regions, cursor shapes, icon shapes, and tile and stipple patterns.

The *bmtoa* and *atobm* filters convert bitmap files to and from ASCII strings. They are most commonly used to quickly print out bitmaps and to generate versions for including in text.

FIG

**xfig**

Facility for Interactive Generation of figures: Menu-driven tool that allows you to draw and manipulate objects interactively in an X window. Requires a three-button mouse.

**f2p**

Translates fig output into pic language.

**f2ps**

Translates xfig output format into the PostScript language.

**fig2dev**

Translates Fig code to various graphics languages.

**transfig**

Creates a makefile to translate figures described in Fig code, PostScript, or PIC into a specified LaTeX graphics language.

X interfaces to other programs

**xman**

Manual page browsing program for the X Window System.

**xmh**

X interface to the MH message handling system. This program has no local support.

**xrn**

X-based interface to the USENET news system that uses the NNTP remote news server for accessing newsgroups and articles.

Display/demo programs

**ico**

Animate an icosahedron or other polyhedron

**maze**

An automated maze program

**plaid**

Paint some plaid-like patterns in an X window
puzzle  15-puzzle game for X
xeyes    Eyes on the screen track your mouse cursor
xgc      X graphics demo
xlogo    X Window System logo
References

[1] Fulton, J.
Using and Specifying X Resources.
1988
Copyright 1988, Massachusetts Institute of Technology.

OSF/Motif Programmer’s Guide.
1989

1990

X Window System User’s Guide: X11 R3 & R4 of the X Window System
1990

Introduction to the SCS Computing Environment.
1997
School of Computer Science
Carnegie Mellon University
Symbols
.mwmrc 42
A
access list 35
active window 17
ALT key
different names for 1
autoraise
distinction from "focus" 40
focusAutoRaise resource 41
B
background image 34
buttons
full-zoom 16
iconify 16
maximize 16
minimize 16
C
changing window size 19
class names
of clients 38
of resources 40
client 3
clientDecoration resource 41
command line options 32
creating a window 19
cut and paste 22
cut buffer 23
D
default environment
changing 31
de-iconify window 22
F
focusAutoRaise resource 41
fontList resource 42
fonts
listing 45
specifying 46
viewing 46
full-zoom button 16
G
gallery option
example 34
I
icon manager 16
iconDecoration resource 42
iconify button 16
iconifying a window 21
instance names
  of applications 38
  of resources 40
K
keyboardFocusPolicy resource 40
L
listener window 17
listing fonts 45
M
maximize buttons 16
menus
  defining 44
  root 17
  window 17
minimize button 16
Motif Window Manager
  see mwm
mouse
  defining button functions 44
moving windows 19
mwm 3, 16
  clientDecoration resource 41
  focusAutoRaise resource 41
  fontList resource 42
  iconDecoration resource 42
  keyboardFocusPolicy resource 40
  windowMenu resource 42
.mwmrc 42
P
paste 22
profiles
  standard login 5
  standard X 5
R
raising windows 22
resource database 4, 37
resources 31, 37
root menu 17
root window 6, 15
  background image 34
S
scrollbar 16, 22
  specifying in .Xdefaults file 37
scrolling a window 22
server 3
starting X 31
startup files
  .mwmrc 5