Introduction to the X Window System

Mike Cote, CCS
<mcote@ria.ccs.uwo.ca>

March 19, 1991

Overview

- Background
- X Architecture
- Starting X
- Window Managers
- Other Clients
- X Resources
- X Fonts
- X at CCS

Background

- The X Window System (X for short) is a network-based graphics window system.
- X was developed at MIT Project Athena by Robert Scheifler, et al.
- X was based on W by Paul Asente and Brian Reid at Standord
- ... which was designed as a window interface to the V-system, an experimental network-based operating system.
- Original intent was to provide a windowing system on Project Athena's various hardware. Other goals were to provide the following:
  - vendor-independence for portability of application software

  - network transparency
  - low-level graphics workstaion services: windows, 2D color bitmap graphics, multiprocess access to workstation screen, low-level input processing
  - policy-free windowing system with no particular *look-and-feel*

- Some Milestones...
  - May 1984 - first window system named X
  - February 1985 - Version 6, first licenced version
  - February 1986 - Version 10.3 - first widely ported X implementation
  - March 1986 - UWO's first exposure to X - Computer Science Dept.
  - September 1987 - X11R1, first implementation of current architecture
  - January 1988 - MIT X Consortium founded
  - March 1988 - X11R2, first Xt implementation
  - April 1988 - Open Look interface announced
  - May 1988 - Open Software Foundation founded
  - November 1988 - X11R3, widely ported stable development platform
  - August 1989 - Motif 1.0 released
  - January 1990 - X11R4, improved and up-to-spec
  - March 1990 - Beginning of CCSs commitment to X-windows

- The MIT X Consortium - their goals and role.
  - founded in January 1988, to support and control the development and evolution of X.
  - As of April 1990, there were about 30 members (including names such as Apple, AT&T, CDC, Cray, DEC, IBM, Motorola, Sun, Xerox) and about 50 Affiliates.

- primary goal is to promote cooperation within the computer industry in the creation of standard software interfaces at all layers in the X Window System environment.
- Also, to promote X as an industry standard and urge vendors to participate in its development and definition of interfaces.
- to cooperate with both university and industry concerns in developing the system.
- MIT's role is to provide the vendor-neutral architectural leadership required to keep the Consortium running.

What is X and Why is it so Popular?

- At its lowest level, X is defined as a protocol between clients and servers for window operations
- X is a network-based graphics window system
- Hardware independent (mostly)
- Operating System independent
- Highly customizable

X Architecture Overview

- The X protocol is what makes X different from other window systems
- Most window systems are *kernel-based*, requiring close ties to the operating system

- X is comprised entirely of *user-level* programs
- The X architecture is based upon the *client-server* model
- The *display server* provides display capabilities and keeps track of user input (keyboard and mouse)
- X clients are the application programs that perform specific tasks
- the clients make requests to the server, which are then communicated to the hardware display by the server
- This division allows the clients and the X server to work together either on the same machine, or on different systems connected by a network
- To port X to a new platform, one must only write the hardware specific portions of the server - clients are then easily ported from system to system (in theory)

The X Display

- Consists of a *Root Window* and number of client windows and icons
- An X display is referenced by the hostname, a server number (usually 0), and a screen number (usually 0 or 1). For example:
  - `hadrian.ccs.uwo.ca:0.0` refers to hadrian's display.
  - `shadow.ccs.uwo.ca:0.1` refers to the second screen of shadow's display.
- As this scheme implies, the terms *display* and *screen* are not equivalent. A display may consist of multiple screens.

- The display to be used is usually determined by the **DISPLAY** environment variable, or with the command line option `-display`.
- Display resolution is usually about 1000x1000 pixels (1152x900 on a Sun3)
- Locations on the display are referenced by a *geometry* specification, which is a modified x-y co-ordinate system using pixels as units.

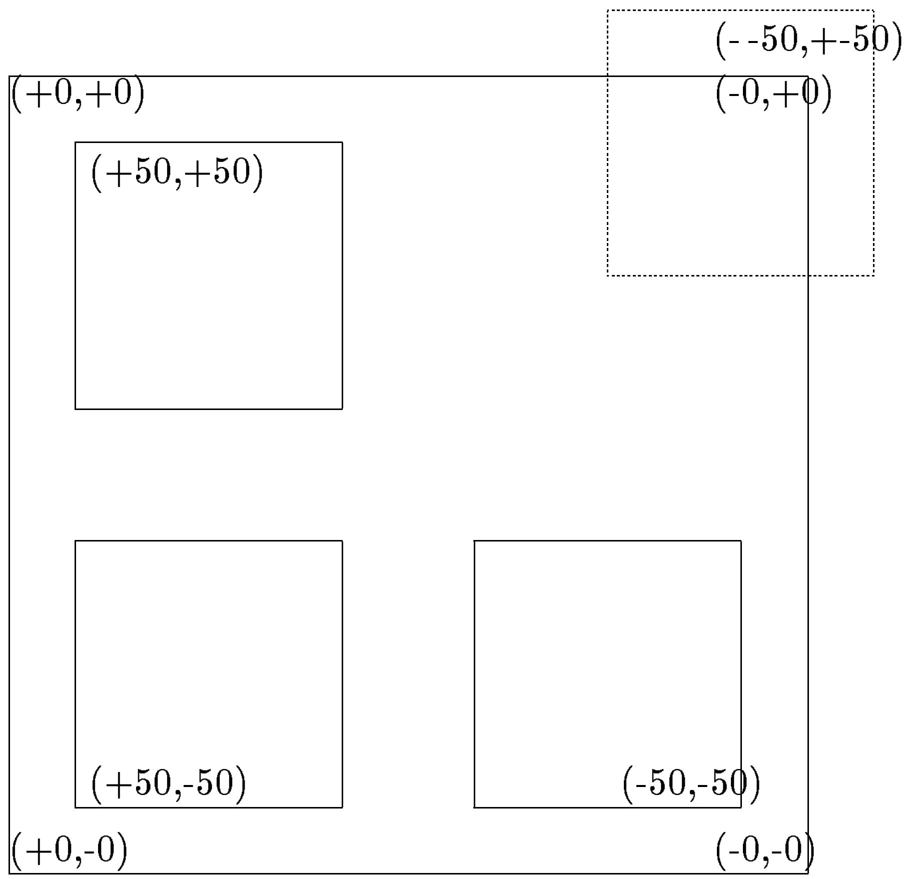


Figure 1: window geometry

- Another part of the *geometry* specification is the width and height of the window. The syntax is: `WIDTHxHEIGHT+XOFF+YOFF`

- A point to be wary of is that the units of the width and height parameters are application dependent. Some examples:

```
xterm -geometry 80x24+30+200 &
xclock -geometry 64x64-0+0 &
```

Starting X

- Currently, there are two different ways of starting X
- xdm - The X Display Manager
  - This is the preferred method of starting X-windows
  - Initially written for X terminals
  - Provides services similar to init, getty, and login on character terminals
  - A workstation running xdm cannot run another window system
  - Xdm reads the file .xsession (if it exists) to determine the user's initial setup
  - A default configuration is provided
- xinit - X Window System Initializer
  - Used before xdm was written
  - X is started from the workstation console
  - Syntax can be complex, so it's usually buried in a shell script
  - xinit uses a different startup file by default (.xinitrc)
  - If no .xinitrc file exists, a single terminal window (xterm) is opened
- User Configuration Files
  - .xsession - an executable script containing the applications to be started.
  - .xdefaults - contains a list of *resources*.
  - .xconsole - only on machine without consoles (ie. Xstations)

- .xbackground - for using a different bitmap for your root window
- .xaccess - other hosts allowed to access your server
- .xmodmaprc - private keyboard remapping
- .xfonts - alternate font path
- .twmrc, .mwmrc, .olwmrc, etc - The window manager configuration file

Window Managers

- X Windows concentrates control in a window manager, rather than in the window system itself, as a kernel-based window system operates.
- The window manager largely determines the *look-and-feel* of the system
- Purpose is to 'manage' windows... A few of the functions generally provided by a window manager are:
  - titlebars and window frames
  - moving and resizing of windows and icons
  - raising and lowering of windows
  - iconifying and deiconifying of windows
  - menus
  - keyboard/mouse accelerators
- Several window managers are available. The most common are:
  - *uwm* - Universal Window Manager (pre R4)
  - *twm* - Tab (Tom's) Window Manager
  - *vtwm*, *tvtwm* - Tom's Virtual Tab Window Manager

  - *mwm* - OSF/Motif^{TM} Window Manager
  - *olwm* - OPENLOOK^{TM} Window Manager

Tab (Tom's) Window Manager

- *twm* is the window manager supplied with the MIT X distribution
- most commonly used here
- *twm* is normally started as the last step of the .xsession (or .xinitrc) script.
- There are MANY ways in which *twm* can be customized. Only a few will be discussed here.
- *twm* window manipulation functions can be invoked in four ways:
  - Using the titlebar
  - Using the menu(s)
  - By combinations of keyboard keys and pointer (mouse) buttons
  - Automatically when a client is started, to allow sizing and positioning of the client window
- The *twm* titlebar normally contains the name of the window, two buttons, and a region that is highlighted when the input focus is in that window.
- The *twm* menu(s) allow access to many of the most frequently used window manipulation functions.
- These menus are usually accessed by pressing and holding down a pointer button while the cursor is on the root window.

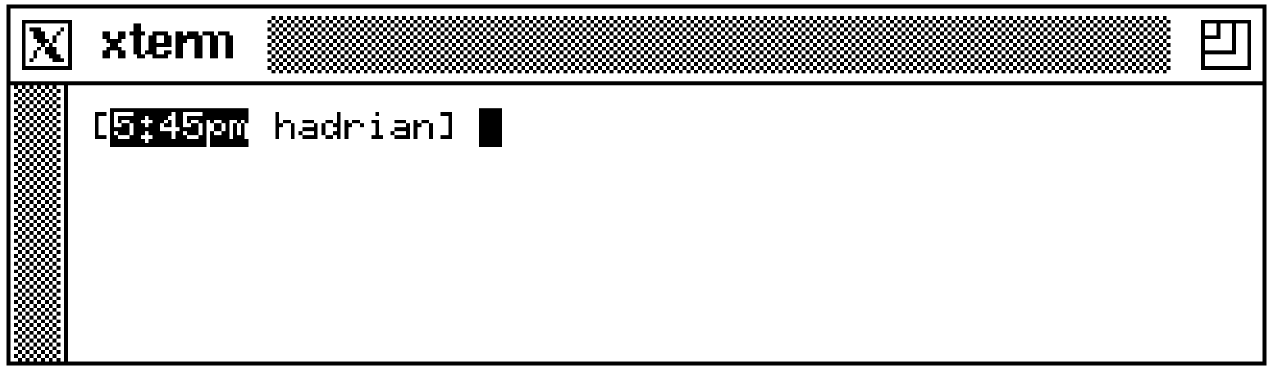


Figure 2: An xterm window with a twm titlebar

- Different menus can be accessed with different mouse buttons.
- The manner of access as well as the contents of the menus are fully customizable via the .twmrc file.
- Any function available on a menu can also be tied to a combination of function keys and pointer buttons.

```
menu "TWM"
{
    "TWM Window Manager"        f.title
    "xterm-24"          !"xterm -sb -sl 500 &"
    "xterm-48"          f.exec "xterm -sb -sl 500 -geom 80x48 &"
    "Window"            f.menu "WINDOW"
    "X Services"        f.menu "SERVICES"
}

menu "titlebar"
{
    "autoraise"         f.autoraise
    "lower"             f.lower
    "zoom"              f.zoom
    "destroy"           f.destroy
    "delete"            f.delete
}
```

Figure 3: Some twm menu definitions

```
Button1 =       : r     : f.menu "APPLICATIONS"
Button2 =       : r     : f.menu "MACHINES"
Button3 =       : r     : f.menu "TWM"
Button1 = m     : r|w   : f.menu "TWM"
Button1 = c m   : w     : f.raise
Button2 = c m   : w     : f.move
Button3 = c m   : w     : f.lower
Button1 = m s   : w     : f.iconify
Button2 = m s   : w     : f.move
Button3 = m s   : w     : f.resize
Button1 =       : t     : f.raise
Button2 =       : t|i   : f.move
Button3 =       : t     : f.menu "titlebar"

Note:  m - meta         r - root window
       c - control      w - window (any other)
       s - shift        t - titlebar
                        i - icon
```

Figure 4: Some function bindings

Some Common Clients

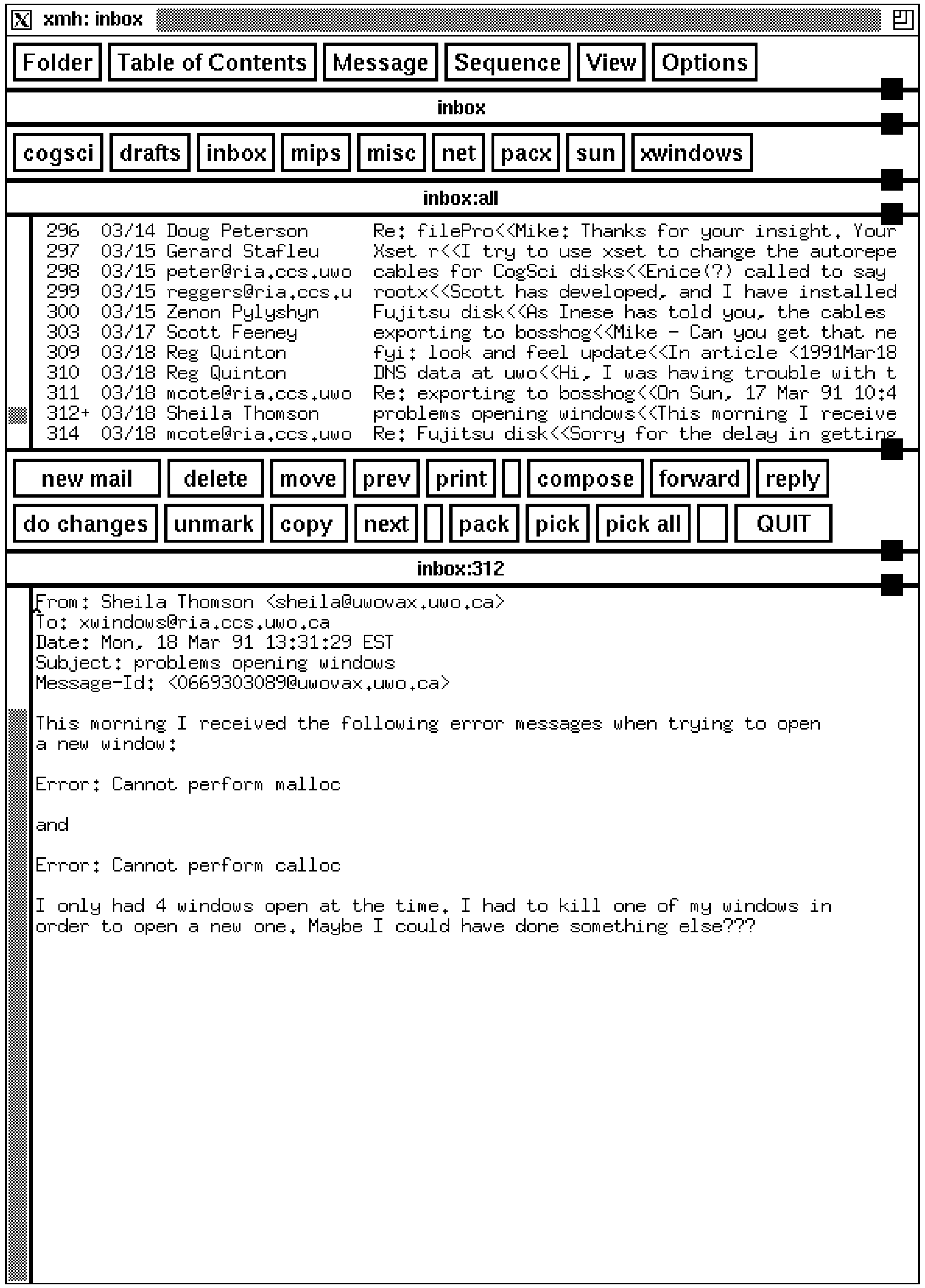
- xterm - emulates a DEC VT102 terminal or a Tektronix 4014 terminal
- xclock - display the time in either digital or analog format
- xcal - A calendar program
- xpostit - postit notes
- xmh - X interface the the *mh* mail system
- xrn - An X news reader
- xfig - A Drawing utility
- xdvi - DVI Previewer for the X Window System
- xps, gs - Postscript previewers
- xcalc - A scientific calculator
- xsetroot - root window parameter setting utility
- xwd, xpr, xdpr, xwud - utilities for dumping, printing and displaying images of an X window
- xfontsel - allows viewing of available fonts
- xlsfonts - lists available fonts
- xrdb - X server resource database utility
- xhost - server access control program for X
- xset - user preference utility for X

The X Resource Database

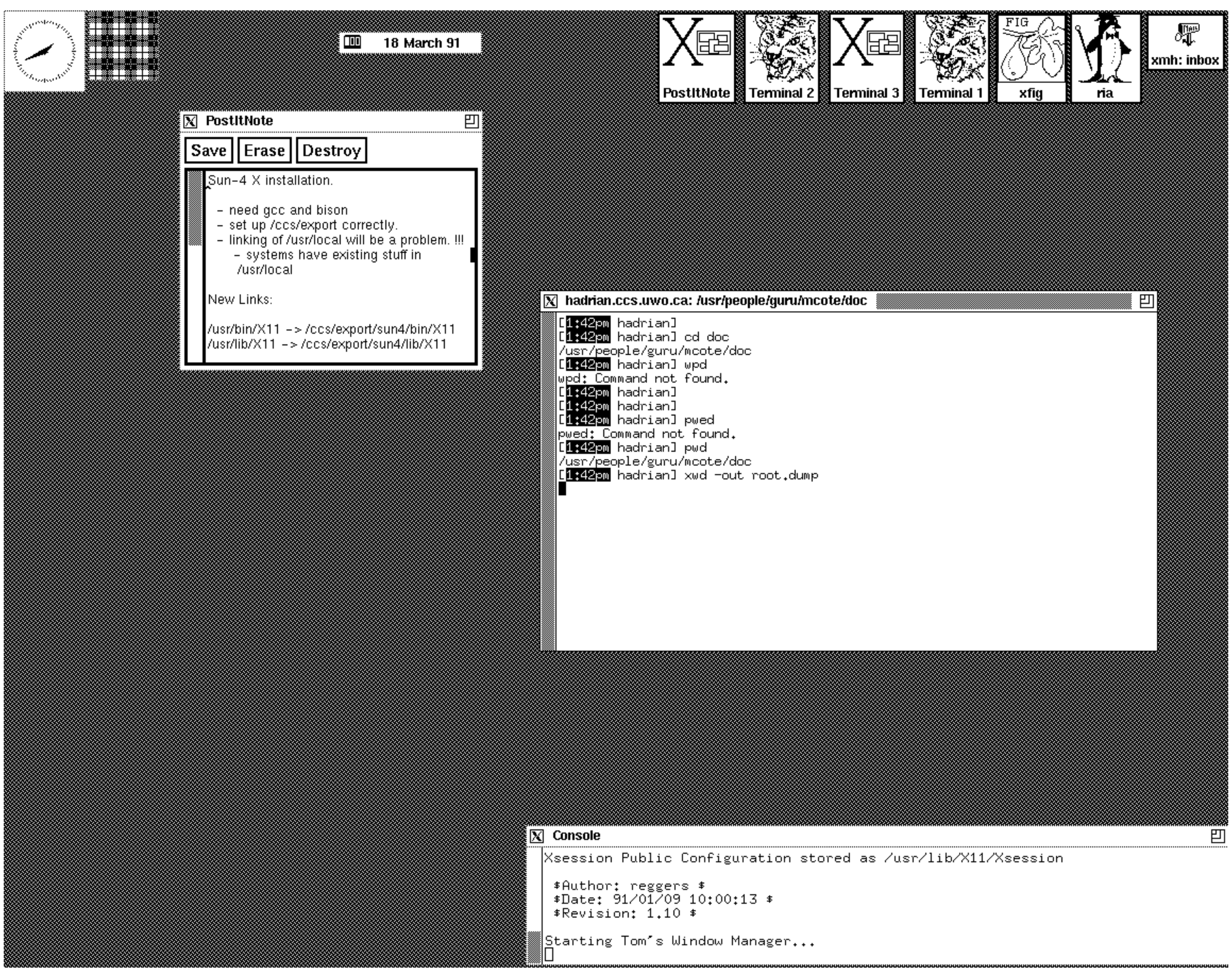
- There are too many options available to an X client to make it practical to use command line arguments.
- Each individual client makes use of a set of default values known to the server. This information is called the *resource database*.
- When the X server is started, the *xrdb* utility is run to initialize the resource database with the contents of the .xdefaults file.
- This information is typically user preferences about color, fonts, etc. on both a global and per client scale.
- In addition to the server resources, each client may (should) have an application resource file of it's own.
- To take it a step further, a user may set the environment variable **XAPPLRESDIR** to the name of a directory containing personal application resource files.
- When a client is started, It's resources are determined by the following procedure:

  1. The current resource variables are obtained from the server.
  2. The system-wide application defaults file for that client is read.
  3. The private application defaults file for that client is read.
  4. Applicable command line arguments are processed.

- As Each step overrides the previous, this allows great flexibility in customization of an application's *look-and-feel*.


xmh: inbox
Folder
Table of Contents
Message
Sequence
View
Options
inbox
cogsci
drafts
inbox
mips
misc
net
pacx
sun
xwindows
inbox:all
296 03/14 Doug Peterson Re: filePro<<Mike: Thanks for your insight. Your
297 03/15 Gerard Stafleu Xset r<<I try to use xset to change the autorepe
298 03/15 peter@ria.ccs.uwo cables for CogSci disks<<Enice(?) called to say
299 03/15 reggers@ria.ccs.u rootx<<Scott has developed, and I have installed
300 03/15 Zenon Pylyshyn Fujitsu disk<<As Inese has told you, the cables
303 03/17 Scott Feeney exporting to bosshog<<Mike - Can you get that ne
309 03/18 Reg Quinton fyi: look and feel update<<In article <1991Mar18
310 03/18 Reg Quinton DNS data at uwo<<Hi, I was having trouble with t
311 03/18 mcote@ria.ccs.uwo Re: exporting to bosshog<<On Sun, 17 Mar 91 10:4
312+ 03/18 Sheila Thomson problems opening windows<<This morning I receive
314 03/18 mcote@ria.ccs.uwo Re: Fujitsu disk<<Sorry for the delay in getting
new mail
delete
move
prev
print
compose
forward
reply
do changes
unmark
copy
next
pack
pick
pick all
QUIT
inbox:312
From: Sheila Thomson <sheila@uwovax.uwo.ca>
To: xwindows@ria.ccs.uwo.ca
Date: Mon, 18 Mar 91 13:31:29 EST
Subject: problems opening windows
Message-Id: <0669303089@uwovax.uwo.ca>
This morning I received the following error messages when trying to open
a new window:
Error: Cannot perform malloc
and
Error: Cannot perform calloc
I only had 4 windows open at the time. I had to kill one of my windows in
order to open a new one. Maybe I could have done something else???

An X Windows Display



- These resource files each contain a list of lines obeying the following syntax:
  object.subobject[.subobject...].attribute: value

  where:

  - *object* is the client program.
  - *subobjects* correspond to major structures within an application, such as windows, menus, scrollbars, buttons, etc.
  - *attribute* is a feature of the last subobject, such as the label of a button.
  - *value* is the setting of the resource attribute.

- An asterisk (*) can be used as a wildcard to create a *loose binding* between components of the specification.

```
*font:                  7x13
*boldFont:              7x13B
xterm*visualBell:       false
xterm*JumpScroll:       on
xterm*scrollBar:        on
xterm*borderwidth:      3

xterm*foreground:      darkblue
xterm*cursorColor:      darkblue
xterm*pointerColor:     darkblue
```

Figure 5: Some X Resource specfications

X Fonts

- Many clients allow you to specify the fonts to be used for different components of the application's windows.
- However, X has a fairly complex font naming system which can be difficult to understand and use.
- While this doesn't sound like it presents much of a problem, here's a list of a few of the fonts available (shown by *xlsfonts*).

```
-adobe-courier-bold-o-normal--10-100-75-75-m-60-iso8859-1
-adobe-courier-bold-o-normal--11-80-100-100-m-60-iso8859-1
-adobe-courier-bold-o-normal--12-120-75-75-m-70-iso8859-1
-adobe-courier-bold-o-normal--14-100-100-100-m-90-iso8859-1
-adobe-courier-bold-o-normal--14-140-75-75-m-90-iso8859-1
-adobe-courier-bold-o-normal--17-120-100-100-m-100-iso8859-1
-adobe-courier-bold-o-normal--18-180-75-75-m-110-iso8859-1
```

- The X font naming system was designed for maximum flexibility rather than ease of use.
- A font name consists of 12 components:
  - *foundry* - The font's developer
  - *font family* - Courier, Helvitica, etc.
  - *weight* - bold, medium, etc.
  - *slant* - oblique, roman, italic, etc.
  - *set width* - normal, condensed, semi-condensed, narrow, double-width
  - *size* - in pixels
  - *size* - in tenths of a point
  - *horizontal resolution* - dots per inch

  - *vertical resolution* - dots per inch
  - *spacing* - monospace (fixed) or proportional (variable)
  - *average width* - in tenths of a pixel
  - *character set* - usually iso8859-1

- as with the resources, an asterisk ($*$) can be used as a wildcard within a font name.

- Common fonts are aliased to more easily used names. This is done in the fonts.alias file in each of the X11 fonts subdirectories.

X Windows at CCS

- As provided, the MIT X distribution comes with a minimal amount of configuration. One of our main efforts has been to come up with a configuration which will allow the novice X user to sit down and start working.
- This has been accomplished, although there are still a few rough edges.
- A complicating factor has been the different hardware platforms currently running X. These are Sun/3, Sun/4, Mips (via NCD X-stations), VAX/VMS, and Silicon Graphics systems.
- With the exception of the VAX, all of these systems are currently running X11 - Release 4 and work well together.
- Another goal has been to make X available for export to the campus in general. This is close to being realized.

The .xsession file

```
#!/bin/sh
#
# $Author: reggers $
# $Date: 91/02/04 06:59:48 $
# $Revision: 1.13 $
# $State: Exp $
# $Locker:  $
# $Source: /usr/src/usr.local/Xlook/RCS/xsession,v $
#
# .xsession is executed once when you start up X. This sets some X defaults,
# starts some programs, and then the window manager. This works on Suns and
# Xstations. This is the system default .xsession (for those users who don't
# have one). If you want to construct a private .xsession you'd usually start
# with this one. If you'd like to preserve the look and feel and just want
# to start some more applications or set some environment variables you
# could have an .xsession that says
#
#   EDITOR=xvi; export EDITOR
#   PRINTER=apple; export PRINTER
#   XMONCONF=/usr/local/lib/snmp/xmon.conf
#   snmpxmon  $XMONCONF &
#   exec /usr/lib/X11/Xsession
#
# Don't forget, if you do this you really need to make sure you
# run your applications in the background!
#
# Original work by Reg Quinton <reggers@ria.ccs.uwo.ca>;
#                  Fri May 11 10:29:40 EDT 1990
# (as originally lifted from Colleen Bretzlaff's area)

# set -x

# Note... you need a search path.... Xdm will set one for
# you that should be adequate, most users like to have add their
# personal stuff in the search path.
```

```
PATH=:/usr/local:/usr/local/bin:/usr/local/share/bin:/usr/X11/new:/usr/X11/bin:/usr/X11/old:/usr/new:/usr/bsd:/usr/bsd43/bin:/usr/ucb:/usr/hosts:/usr/bin:/bin:/usr/lbin:/usr/sbin

# Most user's have a /bin of their own

if [ "$USER" != "root" ]; then
PATH=$HOME/bin:$PATH
fi
export PATH

# There's lots of environment variables I want to honor if already
# set (eg. by .login, or by xdm). One might want to force particular
# settings -- eg. force a particular font path. The notation
# X=${X-Y} means if X is not set, then set it to Y. An awkward
# way of allowing for changes from a parent xdm session.

XLIB=${XLIB-/usr/lib/X11}; export XLIB
BITMAPS=${BITMAPS-/usr/include/X11/bitmaps}; export BITMAPS
XSESSION=${XSESSION-$HOME/.xsession}; export XSESSION
SYS_XSESSION=${SYS_XSESSION-$XLIB/Xsession}; export SYS_XSESSION
XDEFAULTS=${XDEFAULTS-$HOME/.xdefaults}; export XDEFAULTS
SYS_XDEFAULTS=${SYS_XDEFAULTS-$XLIB/Xdefaults}; export SYS_XDEFAULTS
XMODMAP=${XMODMAP-$HOME/.xmodmaprc}; export XMODMAP
SYS_XMODMAP=${SYS_XMODMAP-$XLIB/Xmodmaprc}; export SYS_XMODMAP
FONTS=${FONTS-$HOME/.xfonts}; export FONTS
SYS_FONTS=${SYS_FONTS-$XLIB/Xfonts}; export SYS_FONTS
HOST=`hostname`; export HOST
EDITOR=${EDITOR-emacs};          export EDITOR
if [ "$EDITOR" = "vi" ]; then
EDITOR="xterm +ut -tn xterm -title Vi -name Vi -e vi"
export EDITOR
fi
PRINTER=${PRINTER-draft};         export PRINTER
POSTSCRIPT=${POSTSCRIPT-PostScript}; export POSTSCRIPT
TERM=xterm; export TERM

# We MUST have a DISPLAY environment variable
```

```
if [ "$DISPLAY" = "" ]; then
if tty -s ; then
echo "Display ($HOST:0.0)? \c"; read DISPLAY
fi
if [ "$DISPLAY" = "" ]; then
DISPLAY=$HOST:0.0
fi
export DISPLAY
fi

# Just cut a record

echo Xsession Public Configuration stored as /usr/lib/X11/Xsession
echo
echo ' $Author: reggers $'
echo ' $Date: 91/02/04 06:59:48 $'
echo ' $Revision: 1.13 $'
echo

# Mail is located in lower left corner by the resource manager

if [ -r $HOME/.forward ]; then
echo "You seem to receive mail elsewhere ..."
else
xmh -flag -initial inbox &
fi

# Clock, postit, and calendar are clustered in upper right

xclock -geometry 100x100-8+2 &

# read your calendar .... (if you have one)

if [ -r $HOME/Calendar ]; then
xcal -geometry 200x25-10+118 &
fi
```

```
# read your postit notes ... (if you have any)

if [ -r $HOME/.postitnotes ]; then
xpostit -geometry 100x100-115+2 &
fi

# Terminal Window in about center of screen

xterm -geometry 80x24+300+200 &

#
# start up the window manager, we support Tom's. You can try motif,
# uwm, etc... but don't ask for help
#

if [ -r $HOME/.twmrc ]; then
echo Starting Tom\'s Window Manager...
exec twm
elif [ -r $HOME/.mwmrc ]; then
echo Starting Motif Window Manager...
exec mwm
elif [ -r $XLIB/twm/system.twmrc ]; then
echo Starting Tom\'s Window Manager Public Configuration...
exec twm -f $XLIB/twm/system.twmrc
elif [ -r $XLIB/mwm/system.mwmrc ]; then
echo Starting Motif Window Manager Public Configuration ...
exec mwm
else
echo I don\'t know what to try... trying TWM
exec twm
fi

#
# Oops .... she's really pumping mud to get here
#
```

```
echo Bad news ... nothing works ... go home
exit 1
```

The .twmrc file

```
#!/usr/X11/bin/twm
#
# $Author: reggers $
# $Date: 91/03/18 08:22:11 $
# $Revision: 1.26 $
# $Locker:  $
# $State: Exp $
# $Source: /ccs/export/src/usr.local/Xlook/RCS/twmrc,v $
#
# startup file for Tom's window manager. This defines to a large
# degree the look and feel of things. You should, of course,
# read the manual, but there ought to be enough comments here
# to get you started. Hopefully there's enough here you needn't
# do any customizing.... but that's wishful isn't it.
#
# This *assumes* that twm has been started from Xsession so that
# certain environment variables have been exported into the session,
# namely:
#
# EDITOR ... a window based editor
# DOC ... where is the documentation?
# HOME ... your home directory
# XLIB ... the X library area (/usr/X11/lib)
# SYS_FONTS ... the system font path file (contents is path)
# FONTS ... your font path file (contents is path)
# SYS_XSESSION ... the system session file name ($XLIB/Xsession)
# XSESSION ... your .xsession file name ($HOME/.xsession)
# SYS_XDEFAULTS ... the system defaults file name ($XLIB/Xdefaults)
# XDEFAULTS ... the user's defaults file name (.xdefaults)
# SYS_XMODMAP ... the system key map file ($XLIB/Xmodmaprc)
# XMODMAP ... your key map file ($HOME/.xmodmaprc)
# XHELP ... a helper program
# XGRIPE ... a griper program
# POSTSCRIPT ... the name of a PostScript printer
#
```

```
# Reg Quinton, Thu Dec 13 14:48:05 EST 1990

BorderWidth 4 #biggish borders, for raising.
DecorateTransients #show a title bar on transients to
RestartPreviousState #restore icons when restarting
#ShowIconManager #the icon manager (a menu), show it.
SortIconManager #show them sorted in the icon manager.
NoCaseSensitive #ignore case in icon sort of icon manager
NoGrabServer #don't grab server during menus, etc.
#NoSaveUnders #faster to repaint than to save
#NoTitleFocus #don't focus when pointer moves in.
#OpaqueMove #move the window, not just an outline requires cpu!

#################################################################
# Fonts -- use adobe fonts for buttons and menus (you can't vary the
# fonts, twm uses no resource information!!)

TitleFont       "-*-helvetica-bold-r-normal--17-*-*-*-*-*-*-*"
ResizeFont      "-*-helvetica-bold-r-normal--17-*-*-*-*-*-*-*"
MenuFont        "-*-helvetica-bold-r-normal--14-*-*-*-*-*-*-*"
IconFont        "-*-helvetica-bold-r-normal--14-*-*-*-*-*-*-*"
IconManagerFont "-*-helvetica-bold-r-normal--14-*-*-*-*-*-*-*"

#################################################################
# Color/Monochrome configurations

Monochrome
{
BorderColor "black"
MenuTitleBackground "black"
MenuTitleForeground "white"
}

Color
{
BorderColor "blue"
BorderTileForeground "orchid"
```

```
DefaultBackground "maroon"
DefaultForeground "gray85"
TitleBackground "maroon"
TitleForeground "gray85"
MenuBackground "maroon"
MenuForeground "gray85"
MenuTitleBackground "gray70"
MenuTitleForeground "maroon"
IconBackground "maroon"
IconForeground "CadetBlue"
IconBorderColor "gray85"
IconManagerBackground "maroon"
IconManagerForeground "gray85"
}

####################################################################
# Define some useful functions for motion-based actions. (none used)

# Function "move-or-lower" { f.move f.deltastop f.lower }
# Function "move-or-raise" { f.move f.deltastop f.raise }
# Function "move-or-iconify" { f.move f.deltastop f.iconify }

##################################################################
# Button = MODIFIERS : CONTEXT : FUNCTION
# Where Modifier is -- shift, control, lock, meta, mod1,...mod5
#       Context is -- window, title, icon, root, fram, iconmgr
# ----------------------------------
# In the root window I want applications menus, one is yours to
# customize at will.

Button1 =      : root    : f.menu "APPLICATIONS"
Button2 =      : root    : f.menu "RLOGIN"
Button3 =      : root    : f.menu "CONFIGURATION"

# These are fail safes, when all else fails...

# Button1 = meta : root    : !"$EDITOR $HOME/.twmrc &"
```

```
# Button2 = meta : root    : !"mv $HOME/.twmrc $HOME/.twmrc.old &"
Button3 = shift : root    : !"xterm &"
Button3 = meta : root    : f.twmrc
Button3 = control : root    : f.quit

# In a window the buttons are usually devoted to cut and paste (with XTerm
# some menus with the control-button modifier). This leaves some room to
# assign applications to mouse acctions.e mouse is used with the other
# zoom: fill screen top to bottom (but not left to right)
# hzoom: fill screen left to right (but not top to bottom)
# circleup: move bottom most window to top
# move: without going to title
# circledown: move top most window to bottom
# fullzoom: fill entire screen
# raiselower: like suns L5

Button1 =  shift : window|title|frame  : f.zoom
Button1 =  meta : window|title|frame  : f.fullzoom
Button2 =  shift : window|title|frame  : f.raiselower
Button2 =  meta : window|title|frame  : f.iconify
# Button3 =  meta : window|title  : f.beep
# Button3 =  shift : window|title  : f.beep

# On the window frame/title I want similar behaviour. Usually just
# raise and focus.

Button1 =      : title|frame   : f.raiselower
Button2 =      : title|frame   : f.move
Button3 =      : title|frame   : f.menu "WINDOW"

# Icons are either created, moved, or deicon'd

Button1 =      : icon  : f.iconify
Button2 =      : icon  : f.move
# Button3 =  ....

#################################################################
```

```
# Iconmanager, this could be fancier. But the Iconmanager doesn't
# seem to be too useful a concept anyways...

Button1 = : iconmgr : f.iconify
# Button2 =  ....
# Button3 =  ....

# Function key (sun) to pull up or sink a window
#"L5"    = : t|w|f|i : f.raiselower
# Function key (sun) to iconfy a window
#"L7"    = : t|w|f|i|m : f.iconify

################################################################
# Titlebar buttons:
# You can add whatever you want. Often peopole want one to
# clobber applications but this is tough since some respond to
# delete while others don't; at this stage a menu is the only
# workable solution... to bad

#LeftTitleButton "shine.bm" = f.destroy
#LeftTitleButton "pin_in.bm" = f.delete

#####################################################################
# Icons:
# When you turn things into Icons they're put into an IconRegion
# that gets tiled. Four common choices, use the whole screen
# and paint column or row wise.

Iconregion  "1024x1024+0+0" West  North 100 100 #top-most row (cf DecWindows)
#Iconregion "1024x1024-0+0" North East  100 100 #right-most column (cf NeXT)
#Iconregion "1024x1024+0+0" North West  100 100 #left-most column
#Iconregion "1024x1024+0-0" West  South 100 100 #bottom-most row (cf Motif)

IconDirectory   "/usr/include/X11/bitmaps" # where are icons?
ForceIcons                      #use icons I list
UnknownIcon "ad-x.bm"       #when none said
#WarpCursor                     #zip to deiconfied window
```

```
Zoom            20              #show 20 iconify trace lines

# Icons are stored in /usr/include/X11/bitmps. Use "xshowbitmaps" to check them

Icons           #use these icons for these applications
{
"Console" "console1.bm"
        "calendar"      "caltool.bm"
"Chaplin" "magi.bm"
"cyber" "mickey.bm"
        "emacs"         "textedit.bm"
"Grafxa" "garfield.bm"
"Help" "help.bm"
"Hydra"         "bones.bm"
"Julian" "bear.bm"
"icarus"        "next.bm"
"inbox"         "compose.3D.bm"
"Manual Page" "help.bm"
"Pacx" "pacx.bm"
        "PostItNote"    "note.bm"
"Ria"         "calvin.bm"
"Rlogin" "dpz-rlogin.bm"
        "sc"            "spreadsheet.bm"
"Telnet" "dpz-rlogin.bm"
        "Vi"            "textedit.bm"
"xcal" "caltool1.bm"
"Xman" "help.bm"
"xpostit" "idea.bm"
"xrn" "news.bm"
  "xterm" "cmdtool2.bm"
"Zebra" "hobbes.bm"
}

NoTitle # don't put a title bar on these (mostly panels)
{
"Information"
"calendar"
```

```
"confirm"
"insertFile"
"likeToSave"
"notice"
"pleaseStandBy"
"popup"
"prompt"
"question"
"search"
"xclock"
"xcal"
"xpostit"
"xwatchmail"
"xbiff"
"xload"
}

##########################################################################
#
#  #    #   ####   #    #          #    #  ######  #    #  #    #   ####
#  ##   #  #    #  #    #          ##  ##  #       ##   #  #    #  #
#  # #  #  #    #  #    #          # ## #  #####   # #  #  #    #   ####
#  #  # #  #    #  # ## #          #    #  #       #  # #  #    #       #
#  #   ##  #    #  ##  ##          #    #  #       #   ##  #    #  #    #
#  #    #   ####   #    #          #    #  ######  #    #   ####    ####
#
# Button3 =      : title|frame   : f.menu "WINDOW"
# Play with a particular window...

menu "WINDOW"
{
"Window Operations"     f.title
"Raise/Lower" f.raiselower
"Move" f.move
"Resize" f.resize
"" f.nop
"Vertical (un)Zoom" f.zoom
```

```
"Horizontal (un)Zoom" f.hzoom
"Full (un)Zoom" f.fullzoom
"" f.nop
"Iconify" f.iconify
"Delete" f.delete
"Destroy" f.destroy
"" f.nop
"Autoraise on/off" f.autoraise
"Focus on Pointer" f.unfocus
"Focus on Window" f.focus
}

###################################################################
# Button3 =      : root    : f.menu "CONFIGURATION"
#
# There are a few configuration files and environment variables
# you might want to tinker with ....

menu "CONFIGURATION"
{
"Configuration" f.title
"Version" !"$XINFO 'TWM configuration UWO/CCS $Revision: 1.26 $'&"
"Window Parameters" !"xwininfo | $XHELP - &"
"" f.nop
"Window Server ..." f.menu "ServerConfiguration"
"Window Manager ..." f.menu "TWMConfiguration"
"Resources ..." f.menu "ResourceConfiguration"
"KeyMaps ..."                   f.menu "KeyMapConfiguration"
"FontPaths ..." f.menu "FontPathConfiguration"
"Change Shell ..." f.menu "ShellConfiguration"
"" f.nop
"Source .twmrc" f.twmrc
"Edit .xsession" !"$EDITOR $XSESSION &"
}

# There's lots of Window Server stuff you can do without too much effort.
```

```
menu "ServerConfiguration"
{
"Server Configuration" f.title
"Refresh Screen" f.refresh
"Current Settings" !"xset q | $XHELP - &"
"" f.nop
"Bell Off" !"xset b off"
"Bell Soft" !"xset b 20"
"Bell Default" !"xset b on"
"Bell Loud" !"xset b 100"
"" f.nop
"Keyclick Off" !"xset c off"
"Keyclick On" !"xset c on"
"" f.nop
"KeyRepeat Off" !"xset r off"
"KeyRepeat On" !"xset r on"
"" f.nop
"Mouse Slow" !"xset m 0"
"Mouse Medium" !"xset m 2"
"Mouse Fast" !"xset m 4"
"" f.nop
"Resources ..." f.menu "ResourceConfiguration"
"FontPaths ..." f.menu "FontPathConfiguration"
"KeyMaps ..."                   f.menu "KeyMapConfiguration"
}

# There's lots of Window Manager stuff you can do without too much effort.

menu "TWMConfiguration"
{
"Tom's Window Manger" f.title
"Version" f.version
"Copy Public to Private" !"cp $XLIB/twm/system.twmrc $HOME/.twmrc &"
"Edit Private" !"$EDITOR $HOME/.twmrc &"
"Source .twmrc"      f.twmrc
}
```

```
# you can use a number of shells

menu "ShellConfiguration"
{
"Change Shell" f.title
"Current Shell" !"finger $USER | $XHELP - &"
"" f.nop
"Csh" !"chsh $USER /bin/csh &"
"TCsh" !"chsh $USER /bin/tcsh &"
"Bourne Shell" !"chsh $USER /bin/sh &"
"Korn Shell" !"chsh $USER /bin/ksh &"
"Born Again Shell" !"chsh $USER /bin/bash &"
}


# resources are implemented as a server data base. You load the
# data base using xrdb, applications ask the server for information

menu "ResourceConfiguration"
{
"Resources" f.title
"Current Settings" !"xrdb -query | $XHELP - &"
"Use Private Resources" !"xrdb -load $XDEFAULTS &"
"Use Public Resources" !"xrdb -load $SYS_XDEFAULTS &"
"Create Private" !"cp $SYS_XDEFAULTS $XDEFAULTS &"
"Edit" !"$EDITOR $XDEFAULTS &"
}

# your keyboard is configurable, especially if it doesn't include
# native vt100 support.

menu "KeyMapConfiguration"
{
"KeyMaps" f.title
"Use Private Key Maps" !"xmodmap $XMODMAP &"
"Use Public Key Maps" !"xmodmap $SYS_XMODMAP &"
"Copy Public to Private" !"cp $SYS_XMODMAP $XMODMAP &"
```

```
"Edit" !"$EDITOR $XMODMAP &"
}

# fonts are a big pain. The font tables are pretty sparse so
# finding fonts that work for everyone is tough.

menu "FontPathConfiguration"
{
"FontPaths" f.title
"Browse Fonts" !"xfontsel &"
"Rehash Fonts" !"xset fp rehash"
"Use Default Font Path" !"xset fp default &"
"Use Private Font Path" !"xset fp= `cat $FONTS` &"
"Use Public Font Path" !"xset fp= `cat $SYS_FONTS` &"
"Copy Public to Private" !"cp $SYS_FONTS $FONTS &"
"Edit" !"$EDITOR $FONTS &"
}

##############################################################
# Button1 =      : root    : f.menu "APPLICATIONS"
#
# some common applications people like to use

menu "APPLICATIONS"
{
"X Window Applications" f.title
"Terminal (xterm)" !"xterm +ut -tn xterm -sl 128 -sb -ls -title xterm -name xterm -geometry 80x24 &"
"Mail (xmh)" !"xmh -flag -initial inbox &"
"News (xrn)" !"xrn &"
"Calendar (xcal)" !"xcal -name calendar &"
"PostIt (xpostit)" !"xpostit &"
"PrintWindow (xpr)" !"xwd | xpr -device ps | lpr -P $POSTSCRIPT &"
"Draw (xfig)" !"xfig -P &"
"Manuals (xman)" !"xman  &"
"Edit" !"$EDITOR &"
"Webster"               !"xterm +ut -tn xterm -sl 128 -sb -geometry 80x30 -title Webster -name Webster -geometry 80x30 -e webster &"
"Phonebook"             !"xterm +ut -tn xterm -sl 128 -sb -geometry 80x30 -title Phonebook -name Phonebook -geometry 80x30 -e phonebook &"
```

```
"" f.nop
"Gripe" !"$XGRIPE &"
"Help ..." f.menu "Help"
"Configuration ..." f.menu "CONFIGURATION"
"Remote Connect .." f.menu "RLOGIN"
"" f.nop
"Refresh Screen" f.refresh
"Source .twmrc" f.twmrc
"Lock (xlock)" !"xlock -remote &"
"Logout" f.quit
}

# Help: User's need help on a number of fronts ... this ain't
# perfect (by far!) but will have to do for now
#

menu "Help"
{
"X Window Help" f.title
"Message of the Day" !"$XHELP /etc/motd &"
"NCD Notes" !"$XHELP $DOC/Xncdnotes.txt &"
"X-emacs" !"$XHELP $DOC/X-emacs.doc &"
"X11-Buglist" !"$XHELP $DOC/X11-BUGLIST &"
"X Installation" !"$XHELP $DOC/Xinstall.txt &"
"X Notes" !"$XHELP $DOC/Xnotes.txt &"
"X MH Notes" !"$XHELP $DOC/Xmh.txt &"
"X Look Notes" !"$XHELP $DOC/Xlooknotes.txt &"
"X SGI Notes" !"$XHELP $DOC/Xsginotes.txt &"
"X VMS Notes" !"$XHELP $DOC/Xvmsnotes.txt &"
"X TWM Notes" !"$XHELP $DOC/Xtwmnotes.txt &"
"Unix Intro" !"$XHELP $DOC/intro.unix &"
"Manuals (xman)" !"xman  &"
"" f.nop
"Gripe" !"$XGRIPE &"
}

############################################################################
```

```
# If I've done things right then you likely don't want to edit any of
# the above... you likely WILL WANT TO EDIT what follows, especially the
# set of hosts listed in the RLOGIN menu. Button2 in the root context is
# a good place for you to customize your applications while preserving a
# look and feel that's consistent with everyone else
#
#  ######  #####      #     #####          #    #  ######  #####   ######
#  #       #    #     #       #            #    #  #       #    #  #
#  #####   #    #     #       #            ######  #####   #    #  #####
#  #       #    #     #       #            #    #  #       #####   #
#  #       #    #     #       #            #    #  #       #   #   #
#  ######  #####      #       #            #    #  ######  #    #  ######
#
# Connect to remote machines. We usually like to run the application there,
# not here. This will undoubtedly need to be customized for your favoriate
# machines.
# Button2 =      : root    : f.menu "RLOGIN"

menu "RLOGIN" # you want to edit this don't you?
{
"Remote Connect" f.title
# "Alex" !"xterm +ut -tn xterm -ls -title Alex -name Alex -geometry 80x24 -e rlogin alex.ctrg.rri.uwo.ca &"
# "Chaplin"               !"rsh chaplin.csd.uwo.ca \"xterm +ut -tn xterm -sl 128 -sb -ls -title Chaplin -name Chaplin -geometry 80x24 -display $DISPLAY &\"&"
"Cyber"                 !"xterm +ut -tn xterm -title cyber -name cyber -geometry 80x24 -e telnet nve.uwo.ca &"
"Grafxa"                !"rsh grafxa \"xterm +ut -tn xterm -sl 128 -sb -ls -title Grafxa -title Grafxa -geometry 80x24 -display $DISPLAY &\"&"
"Hydra " !"xterm +ut -tn xterm -ls -title Hydra -name Hydra -geometry 80x24 -e rlogin hydra.uwo.ca &"
# "Icarus" !"xterm +ut -tn xterm -ls -title Icarus -name Icarus -geometry 80x24 -e rlogin icarus &"
"Julian"                !"rsh julian \"xterm +ut -tn xterm -sl 128 -sb -ls -title Julian -name Julian -geometry 80x24 -display $DISPLAY &\"&"
"Ria"                   !"rsh ria \"xterm +ut -tn xterm -sl 128 -sb -ls -title Ria -name Ria -geometry 80x24 -display $DISPLAY &\"&"
"Zebra"                !"rsh zebra \"xterm +ut -tn xterm -sl 128 -sb -ls -title Zebra -name Zebra -geometry 80x24 -display $DISPLAY &\"&"
# "Zoi" !"xterm +ut -tn xterm -ls -title Zoi -name Zoi -title Zoi -geometry 80x24 -e rlogin zoi.lib.uwo.ca &"
"" f.nop
"Pacx96"                !"xterm +ut -tn xterm -sl 128 -sb -title Pacx -name Pacx -geometry 80x24 -e telnet pacx96.ccs.uwo.ca &"
"Rlogin"                !"xterm +ut -tn xterm -sl 128 -sb -title Rlogin -name Rlogin -tn vt100 -e csh -c \"echo -n 'Remote host? '; exec rlogin $< \" &"
"Telnet"                !"xterm +ut -tn xterm -sl 128 -sb -title Telnet -name Telnet -tn vt100 -e csh -c \"echo -n 'Remote host? '; exec telnet $< \" &"
}
```