

R 94 / 39

DL/CSE/TM43

technical memorandum Daresbury Laboratory

DL/CSE/TM43

USER'S GUIDE TO THE DARESBUURY X TERMINAL SERVER

by

R. WHITTINGTON, SERC Daresbury Laboratory

January, 1994

94/28

Science and Engineering Research Council
DARESBUURY LABORATORY
Daresbury, Warrington WA4 4AD

**DARESBUURY
LABORATORY
26 JAN 1994
LIBRARY**

© SCIENCE AND ENGINEERING RESEARCH COUNCIL 1994

Enquiries about copyright and reproduction should be addressed to:-
The Librarian, Daresbury Laboratory, Daresbury, Warrington, WA4 4AD

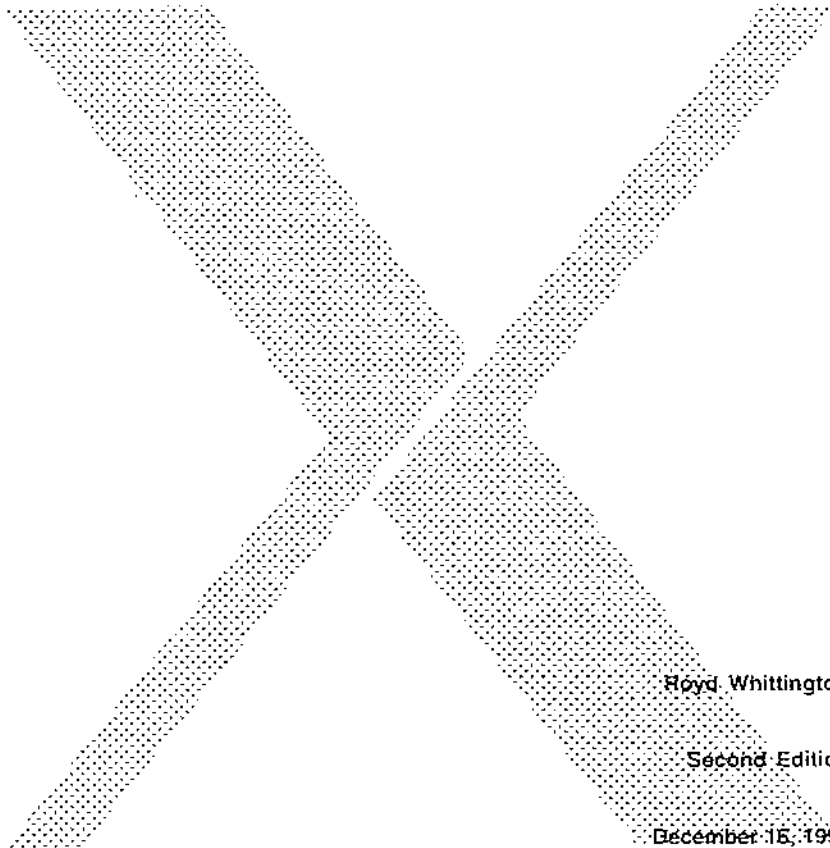
ISSN 0144-5677

IMPORTANT

The SERC does not accept any responsibility for loss or damage arising from the use of information contained in any of its reports or in any communication about its tests or investigations.

SERC Daresbury Laboratory

User's Guide to the Daresbury
X Terminal Server



Royd Whittington

Second Edition

December 15, 1993

Contents

1	Introduction: What is X?	1
1.1	An X Session	1
1.2	Interprocess Communication	2
1.3	Interclient Communication or "Cut and Paste"	3
2	On Your Desk: What is an X Terminal?	4
2.1	CASE STUDY: The Pericom X-line150.	5
3	Behind it all: What is an X Terminal Server?	5
3.1	Functions	5
3.2	xserv1	6
3.3	The Start-up Sequence: what happens when a terminal is switched on?	6
4	The Default Environment: A tour around the standard windows and an introduction to window manipulation.	7
4.1	Window Control	7
4.2	Function Keys and Selections	9
4.3	Menus	9
4.4	Icons	10
4.5	The Workspace Menu	10
4.6	Program Menu	11
4.7	Utility Windows	11
4.8	Terminal Windows	12
4.9	Editors	13
4.10	Drawing Programs	14
4.11	Input Focus	14
5	Customisation: How to set your own style.	14
5.1	Login Customisation — .xlogin	14
5.2	Choosing a Window Manager	15
5.3	Olwm Workspace Properties	15
5.4	xset — User Preference Utility for X	15
5.5	Customising the Workspace Menu	16
5.6	Start-up Windows	16

6	Interacting with other systems	16
6.1	Introduction	16
6.2	Security... and the DISPLAY environment variable	17
6.3	xauth	17
7	The Future: Possible directions and an invitation to comment.	18
Appendix A	Document Scope	19
Appendix B	Bibliography	20
1	Books on X windows	20
2	Unix Man Pages	22
2.1	Using Manual Pages	22
2.2	Useful Manual Pages	22
3	Pericom Terminal Installation	22
4	Software manuals on the Window Environments	22
Appendix D	Known Problems and faults	23
Appendix E	Help and support.	26
1	Mailing List	26
2	Support Groups	26
2.1	Computer Help Desk (UIG)	26
2.2	Operators	26
2.3	Network Support	26
2.4	Postmaster	27
Appendix F	Listings	28

This document describes X terminals, the requirement for an X terminal compute server and the system installed to provide an X terminal service at Daresbury Laboratory¹. As a single source for prospective users of the system, this guide includes, in sections 1 and 2, a brief introduction to X windows. Section 3 describes the X terminal server system. Section 4 describes the default windowing environment and discusses how to modify this to suit your own requirements. Section 5 introduces some of the details of customisation and section 6 discusses using the X terminal with other computer systems. Section 7 discusses future developments and invites you to comment on the service. See Appendix A for a note on the applicability of this document to software and hardware revisions. Other appendices contain a bibliography of X books, a list of some known problems and how to get help and support. The final appendix presents some listings of scripts used in the X terminal server system.

This document uses the following conventions for font usage:

monospaced --- computer filenames or commands

sans-serif --- menu items

boxed --- keyboard keys

1 Introduction: What is X?

X is a software system for interactive computer graphics. X allows you to create *windows* on the terminal screen and responds to mouse and keyboard activity. X performs these tasks with a set of cooperating processes which communicate but otherwise run asynchronously. The language used between these programs is called the *X protocol* and it is this that defines the behaviour of X. X was designed at the Massachusetts Institute of Technology and much useful software is available, free, in source form. The current status of X is version 11 release 5 although most implementations remain at release 4. X has now reached a significant degree of design and implementational stability with no changes to the core X protocol now envisaged and ANSI standardisation is in progress.

1.1 An X Session

The essential programs comprising an X session are an X server, one or more X clients and an optional window manager.

The X server In order to carry out the mechanics of driving the hardware of your terminal, drawing pixels, listening to the keyboard and mouse, and so on, a program called the X server executes with intimate knowledge of this hardware. Invariably this means the server program runs *within* the terminal itself.

The term *server* does demand some clarification. In the area of distributed computing, *server* has come to refer to that hardware or software which directly controls a limited resource. Several examples exist already: file servers control access to a limited resource, namely, files and disks;

¹ The system is managed by Systems Group, Computing and Electronics Division

compute servers control access to the Central Processing Unit (CPU); database servers provide a database resource and X servers allow access to the terminal's screen, keyboard and mouse.

The X server program runs continuously, waiting for external stimuli such as mouse movement or button presses (mouse *events*), keyboard key depressions (keyboard events) or X protocol requests. The X server program is responsible for the physical rendering of line and image graphics and text on the terminal screen. The X server should have available a large family of fonts any of which can be used at any time.

X clients are programs which emit X protocol sequences which are passed using inter-process communication mechanisms such as internal or network sockets to the X server. X clients may also choose to make requests for information from the X server in which case the response is sent back along the same path to the awaiting X client.

One important feature of X windows is that the underlying client/server communication can operate within the same multitasking workstation or in the case of an X terminal, across a local or wide area network. Each X client makes a private connection to the server. Usually X clients connect to just one server during their execution. The X server on the other hand must handle many connections from potentially many X clients.

The X client is responsible for the behaviour of its own windows on the screen. This includes the contents of the window, what happens when the input devices are triggered and what happens to the contents of the window as a result of the input. This can be anything from the behaviour of a button to the control of complex spectrum analysis displays. The appearance of X graphics and the effects of interaction are called the *Look and Feel* of the software.

So, each X client exists in a little world of its own, responsible for its own behaviour only. What controls the way X clients relate to each other? What controls the behaviour when windows overlap, change position or size? This is the domain of the *window manager*.

The window manager is just another X client, albeit a complex one. In X windows, an X client can only **demand** that the contents of the window are as the X client specifies. The relationship of one X client to another is not a matter for any individual client but is left up to the wisdom of the window manager. To do this, the manager registers itself with the X server as a special case and expects to receive notification of certain important events such as creation of new X clients, changes in layout on the screen and input occurring outside of any window.

The window manager may also place useful objects next to existing clients' windows for purposes of moving, resizing and redrawing. The general rule is that the window surrounds are placed there by the window manager which means that although different clients may have different behaviour, the look and feel of the control of all windows is common to all windows²

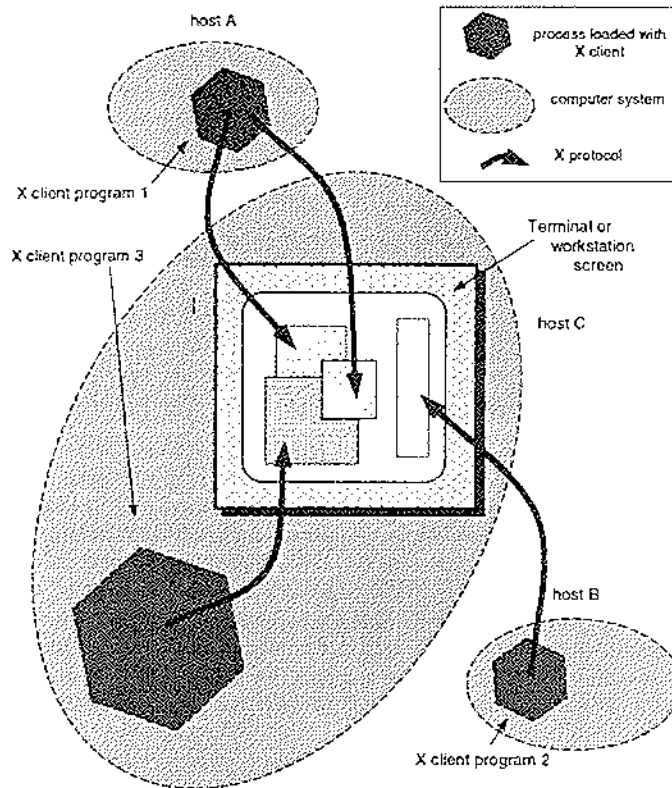
1.2 Interprocess Communication

When the processes involved in an X session are executing on the same UNIX computer system, the underlying mechanism is based on internal UNIX sockets. When communication is across a computer network, as in the case of X terminals, the X protocol is carried by a *general*

² X clients can have many windows, the main ones will have window manager control, but some can be created with a request for them not to be managed and so may have a different surround.

purpose network protocol. In the case of X windows clients on UNIX systems, it is usually the Transmission Control Protocol carried via the Internet Protocol (TCP/IP). These are the same protocols that support many UNIX facilities including the programs ftp(1), telnet(1) and even the spooling software, lpr(1). In the case of other operating systems, other protocols may be used, for example, under VAX/VMS, the DECnet protocols may be used. Figure 1 shows that many connections to the server can coexist.

Figure 1 A single server may drive the screen, accept input from a variety of hardware devices and display windows from many clients. In this figure, X client 3 is shown executing within a workstation computer. An X terminal would not normally support this.



1.3 Interclient Communication or "Cut and Paste"

No facilities are provided within X for communication directly between clients. This is the case because the clients may be on different machines running different operating systems and

network protocols. Clients can however pass information between themselves by using the only common entity known to them all, namely the X server. In this way only the server need support the multifarious means of connection used by the clients.

The most obvious example of interclient communication is "Cut and Paste". Cutting some text from one window and inserting it into another is a useful and popular operation which is supported only by virtue of the clients involved agreeing to abide by conventions laid out in David Rosenthal's Interclient Communication Conventions Manual from MIT X Release 2 which is reprinted in several of the books and manuals mentioned in the bibliography.

The X server provides storage objects called *Properties*. These data areas are set by the *owner* of the selection and can be transferred to a *requestor* client. Often, the owner's selection string is copied and displayed in a requestor's window but how the user interface provides these functions is embodied in the *Look and Feel*. It is possible for new properties to be created which are to contain the names of files or any other data. More information can be found in any detailed X windows guide.

2 On Your Desk: What is an X Terminal?

A typical X terminal looks much like a PC or workstation. What distinguishes them is that the terminal has no Operating System. Otherwise, the hardware is quite similar:

A Screen. X terminals come with all sorts of displays. Monochrome displays have two colours, Off and On. Colour screens have more information at each pixel, i.e., Off and On and shades of colour in between which can be grey levels or other colours, but the number of displayable colours depends on the amount of information that can be stored at each pixel. For example if the terminal has 4 bits of data at each pixel, then 16 colours can be displayed at one time. This type of display is said to have a *depth* of 4. Most colour systems are 8 bits deep.

The Central Processing Unit. X terminals have a CPU, some memory and network interface hardware. In many cases the sole job of the CPU is to load and execute the X server and any support software such as that which provides network protocols. Some terminals are capable of limited local processing for X clients such as a local window manager. This can reduce overall network traffic to and from the X terminal. In a running system, the memory will contain the code for the X server program, copies of parts of the screen, private data storage for the X server program, data in transit to and from the network and unallocated chunks of memory often called fragments.

The Keyboard. The X server is interrupted whenever a key is pressed (called a KeyPress event) and when a key is released (called a KeyRelease event). The X server stores this information and will notify interested clients of these occurrences.

The Mouse. The X server is interrupted when the mouse moves and whenever a mouse button is pressed. The mouse position is always available to X clients.

2.1 CASE STUDY: The Pericom X-line150.

The Pericom X-line150 is one of very many X terminals available from a wide range of manufacturers. The X-line150 is, in fact, a *badge engineered* Samsung SGS-19 X terminal, sold by Pericom in the UK.

The X-line150 has a 19" monochrome screen. The display is 1280 pixels wide and 1024 pixels high and has a resolution of 92 pixels per inch. The monitor and video hardware provides a 66Hz refresh rate.

The CPU is an Advanced Micro Devices AMD29000 running at 16MHz. The terminal is delivered with a default 2 Mbytes main memory but can be upgraded in increments of 1, 2 or 4 Mbyte modules to a maximum of 10 Mbytes. The X server program supports X11 Release 4 and includes set of local menus which display execution parameters such as amount of system memory available.

The keyboard has 101 keys in a PC-like layout and a standard electro-mechanical mouse is included.

3 Behind it all: What is an X Terminal Server?

3.1 Functions

Since an X terminal does not have an Operating System it is not capable of functioning independently. Remote computers are always necessary to execute useful X clients. This dependence on other networked computers leads directly to system management problems and potential performance limitations. It was to assist in monitoring and controlling the operation of a set of X terminals that the X Terminal Server Pilot Project was instigated.

To support the operation of X terminals, a separate computer must provide a number of network services:

1. **Boot service.** The most cost effective X terminals do not even store the X server program in ROM, or permanent memory. The first task for an X terminal server is to detect when an X terminal requesting boot and supply relevant boot programs. System management is located at one source and minimises network related boot problems and should lead to easier and quicker fault diagnosis.
2. **Font service.** X terminals make requests for fonts *on-demand*. There are few prerequisites for the number of fonts provided but any X clients driving any X terminal may request certain fonts. Failure to supply the font at this time can result in run-time failure of the X client program. A rich set of fonts stored on the X terminal server and available at all times is therefore very attractive. A single source for the provision of fonts makes it easier to provide an extensive set of fonts.
3. **Initial Login and Authentication Service.** Before the X terminal can be used as a screen for X clients, some login must occur. The X terminal server provides an initial site for login. This can be followed by logins to other general purpose computers for execution of applications. Authentication is centralised in the X terminal server with obvious benefits in security and user support.

4. **X Windows Environment.** After login, the X terminal server can provide a variety of user environments to suit. This can be configured by individual owners of X terminals but again with a single point of contact in the case of problems.

Along with these benefits are some serious responsibilities. Perhaps the most important is the requirement for *uptime*. Whereas workstations and PC's, with their own Operating Systems can survive failure of central file servers without the loss of the User's environment, loss of the X terminal server results in complete loss of the entire login X windows session. This would normally include any sessions set up to other systems and any current foreground applications.

It should be noted that there is an increased reliance on the network. Whilst workstations, and other desktop machines with Operating Systems on board stand some chance of surviving momentary loss of communication with a host, X terminals are particularly susceptible to network problems.

3.2 xserv1

The system chosen to provide the above network services for X terminals was a SUN Sparcstation 10 Model 41 with 128 Mbytes of memory. The Sparcstation 10 has 64 Kbytes of static RAM cache, 1 Mbyte of secondary cache and can execute 96 MIPS. The system has a single on-board SCSI bus with an internal 204 Mbytes disk for system files and an external 1.2 Gbytes disk for swap (600Mbytes), users' home directories (230Mbytes) and X terminal boot and font disk storage (95Mbytes).

3.3 The Start-up Sequence: what happens when a terminal is switched on?

From power-up³, an X terminal will first perform hardware self checks. The next step is to determine the terminal's own Internet Address, a number which uniquely identifies the terminal both to the network and to the X terminal server itself.

X-line150 terminals use the BOOTP protocol to determine both the IP address and the name of the file on the terminal server which contains that terminal's X server program. Other X terminals may have the X server program in ROM in which case, this and the next step are not required.

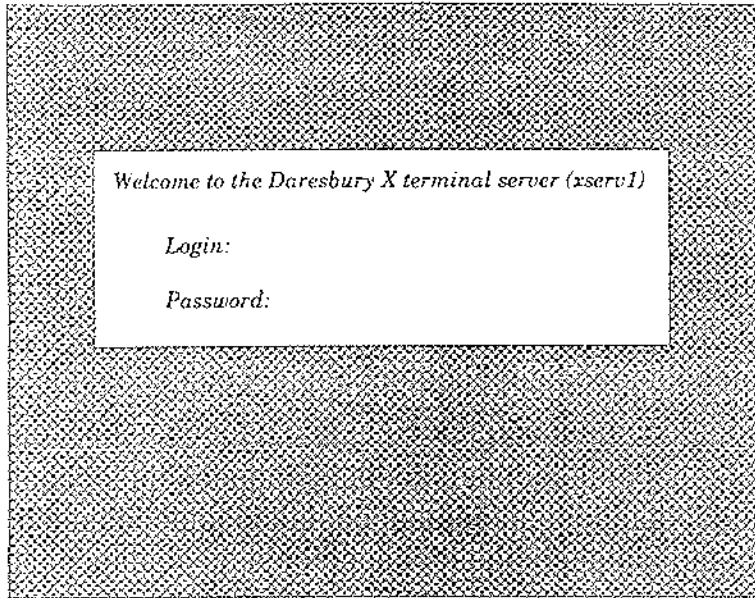
Having determined the location of the X server program file, the TFTP⁴ protocol is used to download the X server program from the X terminal server computer to the terminal. Once complete, initialisation of TCP/IP networking is necessary followed by X operation proper.

X-line150's are configured to use the X Display Manager Protocol (XDMCP). When the X server program starts, a message is passed over the network to the X terminal server to indicate that an X terminal has started. The system (or *daemon*) process in the X terminal server, on receipt of this message, starts an X client program which displays the *authentication window* on the X terminal's screen (see Figure 2). This window prompts for identifier and password.

³ A terminal newly delivered from the factory may have a configuration different from the standard appropriate at DE. In the case of X-line150's a note is available from network support.

⁴ In other configurations, NFS, the Network File System may be the protocol involved.

Figure 2 The authentication window and the prompt for ID and Password.

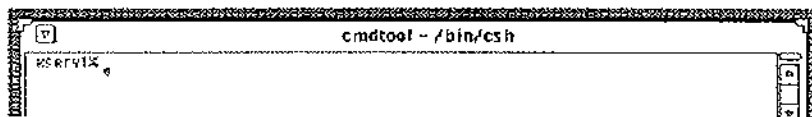


4 The Default Environment: A tour around the standard windows and an introduction to window manipulation.

The default window manager is the OpenLook Window Manager, *olwm* which offers compatibility with Sun workstations. *Olwm* uses a popular convention for the use of mouse buttons, indicated in Table 1. The following sections describe the behaviour of windows using *olwm*. Other X window managers may have different behaviour.

4.1 Window Control

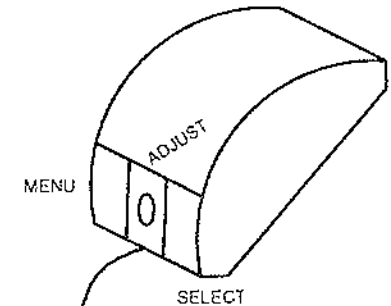
Control of windows is normally done by positioning the mouse over sensitive areas and either using the SELECT or MENU buttons. For overall window control, the best area to use is the *top bar*:




The following actions can be performed using mouse buttons on the top bar.


Table 1 Uses for the three mouse buttons. Some X terminals such as PCs for instance may have a mouse with only one or two buttons. In this case the missing buttons are often available by pressing the CONTROL or Ctr key at the same time as pressing a mouse button (e.g. Ctr + Left Button = Right Button). See the documentation on the PC software for more information.

LEFT	SELECT	Select items; the current window for input, energize buttons, select the default item of menus, move windows about and bring them to the front of the stack of windows, resizing windows, opening icons.
MIDDLE	ADJUST	Adjust prior to completion
RIGHT	MENU	All menus (with a few exceptions, e.g. xterm). Menu dependent on position of mouse pointer. Over background: root menu, Over Window border: window control. Over text area: text editing control.



Moving Windows Press SELECT anywhere on the top bar (except the *menu glyph*: ) and hold. The window remains in place but a *rubber band* outline moves with the mouse pointer to the desired position.

Iconising Windows Press SELECT on the menu glyph. The window will vanish and an icon will appear normally at the edge of the screen. See below for information on Icons.

Resizing Windows Press SELECT on a resize corner:  and pull out the line to the desired size for the window. The whole window will normally be resized after the mouse button is released.

Making Windows Visible Sometimes window are partially or completely obscured by others. Press SELECT on the top bar as if you were moving it but release immediately. The window will pop up above all others.

Window Menu There is also a menu associated with the top bar. Press MENU. See the section below for more details.

Pushpins Some windows have pushpins:  which can be pulled out with the SELECT button to dismiss the window.

4.2 Function Keys and Selections

The system provides some useful defaults for the row of function keys on an Xline 150 and 200 keyboard. This is done via the `Xsession` file, see page 15.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
help	cut	copy	paste	find	undo	again	back	icon	not used	not used	not used

help display on-line help.

cut cut out the current selection and place into the selection holder.

copy copy the current selection into the selection holder.

paste insert the contents of the selection holder into the position immediately after the text pointer.

find find the next instance of the selected text, or the selection holder if no text is selected.

undo undo the last operations back to the last mouse movement.

again redo all operations since the last mouse movement.

back put the window to the back of the stack of windows on the screen.

icon iconise the window if open, or open up to a window if already iconised.

Text and files in the File Manager, can be cut and pasted around windows and from one window to another by using the **selection service** provided by interclient communication (see 1.3). Text is *selected* by using the SELECT button and moving the mouse pointer, highlighting an area of text. This text can be deleted using the **cut** function. The cut text is also saved into the *cut buffer*, an X server property, and can be inserted back into the same or a different window by using the *paste* key. The *copy* key copies into the selection holder without deletion.

4.3 Menus

Menus are used throughout the OpenLook user interface. OpenLook menus can be driven in two distinct ways which have subtly different manipulation of mouse buttons but achieve the same results. For instance, to select the **open** item from an icon's menu (see below) you may follow either of the following sequences:

1. Position the mouse over the icon and *single-click* the MENU button without moving the mouse. A single-click is a depression and release of a mouse button without moving the mouse. You will see the menu remains on the screen even though you do not now have

a button depressed. Position the mouse over the open item and single-click and the icon will open. Single-click the MENU button while the mouse is anywhere else the menu will cancel the operation.

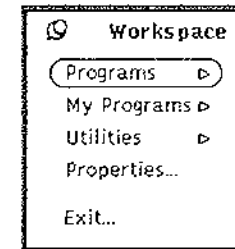
2. Position the mouse over the icon and press and hold the MENU button. You will see the menu appear and stay on the screen. Move the mouse away from the menu and release selects nothing and cancels the operation, but position over the open item, release, and the icon opens up.

4.4 Icons

Icons are created and placed by the window manager as a reminder that a program is running and has one or more windows present. Icons can be *opened* up to reveal the application's window by *double-clicking* on the icon. A double-click is two single-clicks in rapid succession. How rapidly is necessary depends on your Properties... setup, see page 15. Alternatively, you can hit the mouse MENU button for the icon menu. Icons can be moved around the screen by holding the SELECT button and moving the mouse to the desired position.

See the `olwm(1)` for full details.

4.5 The Workspace Menu



Many programs can be started from this main menu. Since the screen background area is known as the *Workspace*, the menu is often referred to as the workspace menu. The various applications are started from the Programs item. The Utilities item contains windowing functions including,

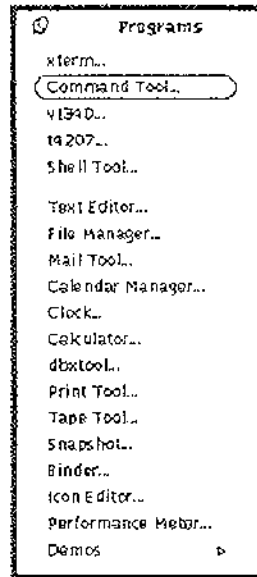
refresh Should it become necessary, this item will ask all X clients to repaint their windows.

save workspace See page 16.

lock screen If you are leaving your terminal and do not wish to log out and so lose your window layout, but do not want to allow others to use your terminal, this item will put the terminal into a secure state but leaving all the applications running. You will need to enter your password to return to the original state.

security The X server has some rudimentary security control. For a remote computer to be able to connect to the X server, the system's name must appear in the X server allowed hosts list. This item starts a client which allows this list to be viewed and changed or switched off completely. More is said about the purpose of this in section 6.2.

4.6 Program Menu



The program menu contains a list of useful utilities. Some are discussed below and manual pages give more complete information.

4.7 Utility Windows

The default environment is shown in Figure 3. The behaviour of these and some other Workspace menu items are discussed below:

Command Tool or **cmdtool** — a basic terminal emulation window. Most UNIX shell commands can be issued here by arranging for *input focus* by positioning the mouse cursor, single-clicking and then typing the commands. See section 4.11 for more information in input focus.

File Manager — a simple file management tool. You can perform all the usual operations on files, or groups of files in the following ways:

- **rename** — use the SELECT button to highlight and edit the file's name directly.

- **move** — files can be moved around directories by opening up the upper part of the display into a *tree* of directories. Do this by using the window menu associated with the upper display. SELECT the file you want to move but **keep the mouse button depressed**. When you move the mouse, you'll find that the file's icon appears to move with the pointer. This is called *drag and drop*. Drag the icon over the directory icon desired *and drop* the file into place.
- **copy** — very much like **move**, except that you must keep the **Ctrl** key depressed at the same time as the drag and drop.
- **delete** — simply **move** the file into the wastebasket!

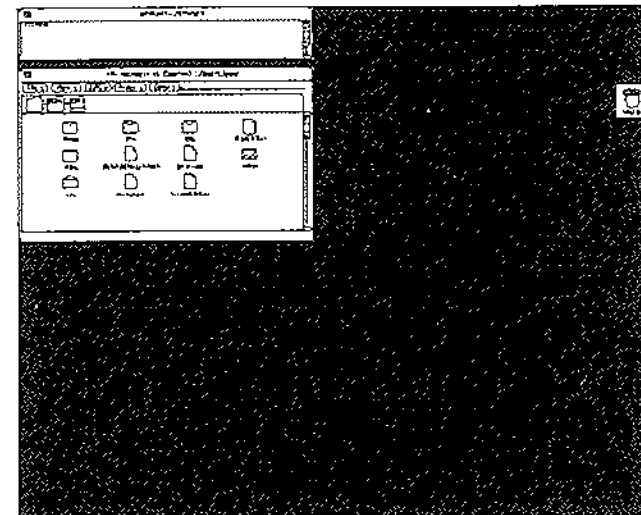
Clock — showing current wall time. The system will adjust for local daylight saving time automatically.

Mail Tool — A convenient user interface to electronic mail. Send a message to *postmaster@dl.ac.uk* if you wish to *receive* mail in this way. You can *send* e-mail immediately.

Calculator — a scientific and programmer's calculator

Snapshot — for taking screen and window dumps and printing.

Figure 3 The default startup window environment.



4.8 Terminal Windows

At first, you are effectively *logged in* to the X terminal server. `ls` will reveal a home directory and disk space which is currently limited, by quota, to 1 Mbyte. If not present already, a symbolic link can be used to access other computer filesystems via NFS. Remember that with NFS, the

Network File System, you can look at and modify files on other computers using local editors (see below) as if they were local files. By convention, filesystems on other computers are available via the name:

Intshostnameldirectoryname

e.g. */nfs/cxa/mnt1*, etc.

You can also log in to remote systems using these windows. Which terminal window type you use depends on what you need and what terminal functions the software you wish to use requires. The following terminal emulators are provided:

Command Tool --- is the usual window in that it supports the same cut and paste operations as most other applications (see page 9). The scroll bar provides history information for very many lines. Find and edit operations are available within the terminal window which are identical to the facilities found in the **textedit** editor.

xterm --- is part of the MIT X windows distribution, provides ANSI standard and Tektronix 4014 emulation and is very commonly used.

vt340 --- provides DEC VT340 functions (which includes VT100 and ANSI X3.64) suitable for EDT and other editors.

t4207 --- provides an emulation of the Tektronix 4207 terminal with ANSI X3.64 terminal control in one window and Tek graphics in another. This window is good for programs such as PLOTEK and EXCURVE.

Shelltool --- This terminal looks like a Command Tool without the scrollbar. In fact, the terminal control sequences are slightly different and this will work when attempting to log into a VMS based machine.

4.9 Editors

Window based editors include:

gemacs --- The GNU version of the emacs system with support for X windows. The is a very sophisticated editor with language support and includes e-mail readers, directory displays and other facilities programmed. Emacs is supported on many different Daresbury computers.

textedit --- The Sun OpenWindows text editor. The function keys are used with this editor, see page 9 for details.

xedit --- The widely available X11 Release 4 text editor.

4.10 Drawing Programs

IslandDraw --- Part of *The Publisher* software suite, this versatile object oriented drawing program is accessible via the commands:

```
package publisher
IslandDraw
```

The program includes on-line help. A notable feature of this program is that it can *read* PostScript files for editing.

xfig --- A Free Software drawing program which has a feature for writing out graphics file suitable for inclusion in LaTeX documents.

IslandPaint --- This *Painting* program treat the image as a region of pixels which can be manipulated directly. No object based operations are possible but selective erasure and pattern drawing are well supported.

xpaint --- A free software item which offers painting features.

4.11 Input Focus

There may be, at any one time, several windows on the screen which offer the ability for you to type text. For example, many utility windows, terminal windows and drawing programs provide text entry areas either for plain text, commands or programming. The problem is that you would normally only want to type in one window at a time but the system doesn't know which one! The problem is solved by window managers which allow you to specify which typing window you wish to receive the characters typed. The default behaviour on X terminals is *click-to-type*. A window is chosen to receive future typing by single-clicking in that window. All subsequent typing will be directed to this X client which, if it is expecting this type of input, will most likely display the characters as they are typed. This window is then deemed to have input focus which can be changed simply by single-clicking in another window.

The default window manager, *otwm*, allows for an alternative mode of working where the single-click is not necessary; the input focus is changed just by moving the mouse pointer over the chosen window. Use the Properties... menu item on the workspace menu to change this. Properties... is described in section 15.

5 Customisation: How to set your own style.

5.1 Login Customisation — .xlogin

At first login, after authorisation, the XDMCP daemon executes the script, */usr/xserv/X11/lib/xdm/Xsession* (see Listing 1 in Appendix F). During this procedure, the system will look for and if possible execute a file called *.xlogin* in your home directory. If you need to set any *environment variables* for the session then they can be set in this file. One problem is that the file **must** be written in the Bourne Shell. (You may wish to use one of the files mentioned below as a template).

5.2 Choosing a Window Manager

One of the tasks of the `xsession` file is to select the window manager from the provided set. However, if the file `.xsession` exists in your home directory then it is executed in place of the system default. Several `xsession` files have been prepared which, if used instead of the default, start different window managers:

<code>/usr/xserv/X11/lib/xdm/olwm_environment</code>	an olwm environment (the default)
<code>/usr/xserv/X11/lib/xdm/twm_environment</code>	a twm environment
<code>/usr/xserv/X11/lib/xdm/olvwm_environment</code>	an olvwm environment

To try these out, copy the files into your own home directory: e.g.

```
cp /usr/xserrv/X11/lib/xdm/twm_environment ~/.xsession
```

then logout and log back in again. To get back to the default environment simply delete your own copy of the `.xsession` file.

5.3 Olwm Workspace Properties

For olwm, the `properties...` item of the workspace menu allows customisation of the following window manager functions:

- Icons — control over icon position on the screen.
- Menus — controls behaviour of menus.
- Miscellaneous — controls frequency of bleeping, input focus control and scrollbar placement.
- Mouse Settings — controls various parameters in mouse operation.
- Localization — where supported, provides control over the language used in user dialogue and how time and numbers are formatted.

In each case, the changes to each setting are only made when the apply button is used.

5.4 xset — User Preference Utility for X

The `xset` command allows you to set various X server parameters:

- Bell or beep on/off, volume and frequency
- Keyclicks on/off and volume
- Search path for fonts
- Keyboard LED displays (usually the Num Lock, Caps Lock and Scroll lock lights)
- Mouse acceleration
- Colour values where possible
- Key auto repeat
- Screen saver

For example, mouse acceleration controls the behaviour of the mouse pointer with respect to movements of the mouse itself. The command:

```
xset m 10/1 10
```

arranges for the mouse to move smoothly until 10 pixels have been traversed in a “short time”. After this the mouse moves faster by a factor of $\frac{10}{1}$. This allows a “flick” of the wrist to move the pointer large distances across the screen quickly. See `xset(1)` for more details.

5.5 Customising the Workspace Menu

`/usr/openwin/lib/openwin-menu` is used to configure olwm unless you have your own file `.openwin-menu` in your home directory which is used instead. The format of these files is described fully in the manual page on olwm. Listing 2 shows the system provided `openwin-menu` file. Notice that this file refers to two other system provided files, shown in Listings 4 and 5.

Customisation

Take a copy of the system default file `/usr/openwin/lib/openwin-menu` and add your own entries, including if you like, a new MENU of your own *specials*. This approach means that new system programs and utilities added by the system administrator are picked up along with your own customisations. See an example of this in Listing 3.

5.6 Start-up Windows

When you log in to an X terminal a default set of windows appear. These windows can be changed or deleted and further windows created by using the Workspace menu or from the command lines of terminal windows. If you arrive at a position that you wish to be duplicated next time you log in you can save this position using the `Save Workspace` menu item under `Utilities`. This option causes a file called `.openwin-init` to be overwritten with most of the correct entries. One caveat is that, for reasons beyond the scope of this guide, some windows may not be represented in the `.openwin-init` file after saving the workspace. You are free to modify the file with an editor, adding more items by hand but beware that doing a `Save Workspace` again will lose these changes.

6 Interacting with other systems

6.1 Introduction

Figure 1 shows how some X clients execute remotely and display their windows on your X terminal screen. Most programs mentioned so far in this document are available on `xserv1` and no *special* actions are required to set them up since the XDM login does this automatically. Also in “Terminal Windows” in section 4.8 it was shown how `xserv1` could *see* files on other systems using NFS. A powerful feature of your X terminal is the ability to display windows from other computers running X clients over TCP/IP. This allows you to login to a remote computer, (not `xserv1`) and display windows on your X terminal in the same way that `xserv1` based programs do. Reasons for doing this might include

1. The required program is not available on `xserv1`.

2. It is a *Science Application*. These are strongly discouraged on `xserv1` so as not to degrade the performance for other users.
3. There is some performance or other benefit to be gained from accessing local disks on the remote system.
4. NFS is not available on the remote system.

6.2 Security... and the DISPLAY environment variable

Two steps need to be taken to setup interaction with other systems.

1. Allow access to your X terminal via the `Security...` item under the `Utilities` menu by specifying the remote machine name via `Add Host` or use `xauth`, see below.
2. Find the name of your X terminal and login to the remote system using (normally) `rlogin`. To determine the name of your X terminal use the command `printenv DISPLAY`. Let's assume your terminal is called "xterm4".

Then, when you have logged into the remote system, type something like:

```
setenv DISPLAY xterm4:0
```

Following these commands, the systems are in a position to execute an X client and display on your X terminal screen.

6.3 xauth

If you are concerned about the security of your X session, bearing in mind that it is possible to *snoop* on the keystrokes of an X session, there is another more sophisticated security mechanism available. Every time you *login* to the X terminal, a security number is allocated to that session and recorded in a file in your home directory (`~/.Xauthority`). The program, `xauth`, controls and displays the contents of this binary file. Access can be allowed to just a single user on a remote system (rather than all users on that system). To allow access from a session on another computer, first display the current authorisation string:

1. Using `xauth`, use the `list` command. The output will look something like this:

```
xauth> list
xauth> xterm4:0 MIT-MAGIC-COOKIE-1 48544639533171563256444741706634
```

2. This is plain text and can be cut and pasted, mailed or copied to the remote system.
3. On the remote system, use the `xauth` command and add the new authorisation string for your X terminal:

```
xauth>add xterm4:0 MIT-MAGIC-COOKIE-1 48544639533171563256444741706634
```

Once done, this remote session can start X programs on your X terminal, but only for the duration of your session on the X terminal. Log out and back in again and the authorisation will fail.

7 The Future: Possible directions and an invitation to comment.

The provision of X terminal services started as a pilot project. Since then, the host system, `xserv1`, has been upgraded in memory and processor speed and is expected to provide X terminal services for up to 40 X terminals and perhaps another 30 login sessions. Performance of the system is monitored both qualitatively and quantitatively and use of network resources, physical memory and typical response times are all watched closely.

The software provided at the start is mostly of Sun Microsystems origin. An environment based on OSF/Motif is also provided. Public domain programs are installed as they become available and upgrades to existing systems will continue.

You are invited to comment at any time on any aspect of the X terminal, the X terminal server, the software and service by sending e-mail to comments@dl.ac.uk

Appendix A Document Scope

This is the First Edition of the Guide is at Version 1.0 and is appropriate for following software and hardware revisions:

xserver1 running xdm (X11R4), OpenWindows V2.0 and V3.0

X-line150's running version Boot 1.3-2, Server 1.3-2.

X-line200's running version Boot 1.3-2, Server 1.3-2

Appendix B Bibliography

B.1 Books on X windows

Title	Authors	ISBN
Distributed Window Systems Featuring X11 and OpenWindows	Allan Davison, Kieron Drake Mel Slater William Roberts	0 201 41637 9
The X Window System A User's Guide	Niall Mansfield	0 201 54438 5
The X Window System An Overview	Niall Mansfield	0 201 56512 9
X Window System Technical Reference	Steven Mikes	0 201 52370 1
X Windows Program Design and Development	Steven Mikes	0 201 55077 6
OSF/Motif Concepts and Programming	Thomas Berlage	0 201 55792 4
Visual Design with OSF/Motif	Shiz Kohara	0 201 56320 1
Mastering OSF/Motif Widgets	Don McMinds	0 201 56342 8
OSF/Motif Programmer's Reference	Open Software Foundation	0 13 640681 5
OSF/Motif Programmer's Guide	Open Software Foundation	0 13 640673 4
OSF/Motif Style Guide	Open Software Foundation	0 13 640616 5
OSF/Motif User's Guide	Open Software Foundation	0 13 640509 6
Open Look Graphical User Interface Application Style Guidelines	Sun Microsystems Inc. AT&T	0 201 52364 7
Open Look Graphical User Interface Functional Specification	Sun Microsystems Inc. AT&T	0 201 52365 5
X Window System Series O'Reilly Books		
Volume 0 X Protocol Reference Manual	Robert Scheifler	0 937175 50 1

Title	Authors	ISBN
Volume 0 X Protocol Reference Manual Release 5	Robert Scheifler	1 56592 008 2
Volume 1 Xlib Programming Manual Release 5	Adrian Nye	1 56592 00 23
Volume 2 Xlib Reference Manual	Edited by Adrian Nye	1 56592 006 6
Volume 3 X Window System User's Guide	Valerie Quercia Tim O'Reilly	0 937175 36 6
Volume 3 X Window System User's Guide Open Look Edition	Ian Darwin	0 937175 81 1
Volume 3 X Window System User's Guide Open Look Edition	Valerie Quercia Tim O'Reilly	0 937175 61 7
Volume 4 X Toolkit Intrinsic Programming Manual	Adrian Nye Tim O'Reilly	0 56592 003 1
Volume 4 X Toolkit Intrinsic Programming Manual Motif Edition	Adrian Nye Tim O'Reilly	0 937175 62 5
Volume 5 X Toolkit Intrinsic Reference Manual	O'Reilly and Associates	1 56592 00 0
Volume 6 Motif Programming Manual	Dan Heller	0 937175 70 6
Volume 7 XView Programming Manual	Dan Heller	0 937175 88 9
Companion to Volume 7 XView Reference Manual For XView Version 3	O'Reilly and Associates, Inc.	0 937175 88 9
Programmer's Supplement for Release 5	O'Reilly and Associates, Inc.	0 937175 86 2
The X Window System in a Nutshell	Edited by Tim O'Reilly Danial Gilly	0 937175 24 X

Title	Authors	ISBN
Color Management System	Al Tabayoyon Joan Taylor Chuck Adams	0 937175 94 3
The X Resource A Practical Journal of the X Window System Issues 0-4	Edited by Adrian Nye	

B.2 Unix Man Pages

B.2.1 Using Manual Pages

Unix on-line manual pages can be read using the `man` command. Below and throughout this document the convention is to refer to the name and the category as follows:

subject(volume)

e.g. `olwm(1)` means the man page for `olwm` from volume 1 which can be accessed with the command:

`man 1 olwm`

If this man page is the only one present in the manual pages then `man olwm` is sufficient.

B.2.2 Useful Manual Pages

See also: `olwm(1)`, `X(1)`, `xterm(1)`, `cmdtool(1)`, `shelltool(1)`, `textedit(1)`, `filemgr(1)`, `mail-tool(1)`, `cm(1)`, `clock(1)`, `calctool(1)`, `dbxtool(1)`, `printtool(1)`, `tapetool(1)`, `snapshot(1)`, `binder(1)`, `iconedit(1)`, `perfimeter(1)` and most of the programs to be found in `/usr/openwin/bin` and `/usr/openwin/bin/xview`

B.3 Pericom Terminal Installation

A short document called "PERICOM XLINE150 X-terminal — Installation Notes [xserv1] (DRAIT D)" is available. This document would normally be used by Systems Group, but should you require a copy ask at the Computer Help Desk.

B.4 Software manuals on the Window Environments

The following manuals are supplied with installed software. They are not purchasable separately and can be obtained for loan via UIG.

1. Sun Microsystems manuals: DeskSet Reference Guide, DeskSet User's Guide.
2. DECwindows Companion to the OSF/Motif Style Guide
3. DECwindows Extensions to Motif — UNIX Systems Edition

E.2.4 Postmaster

If you have electronic mail problems send for help to:

e-mail: postmaster@dl.ac.uk

(This may sound paradoxical; but local mail will mostly work. If even this fails then use network support will be able to help)

Appendix F Listings

Listing 1 /usr/xserv/X11/lib/xdm/Xsession

```
#!/bin/sh
OPENWINDHOME=/usr/openwin
PATH=/usr/xserv/X11/bin:$OPENWINDHOME/bin/xview:$OPENWINDHOME/bin:\
/usr/local/bin/xview:/usr/local/bin:/usr/local/bin/X11:$PATH

LD_LIBRARY_PATH=$OPENWINDHOME/lib:/usr/lib

NOTIFHOME=/nfs/dlab/user/motif1.6

MANPATH=$OPENWINDHOME/man:/usr/man:/nfs/dlab/user/share/man

export PATH OPENWINDHOME LD_LIBRARY_PATH MANPATH NOTIFHOME

case $# in
1)
    case $1 in
    follow|*)
        exec xterm
        ;;
    *)
        exec
    esac
esac

if [ -x $HOME/.xlogin ]; then
    . $HOME/.xlogin # home startup
fi

startup=$HOME/.xsession
resources=$HOME/.Xdefaults

if [ -x $startup ]; then
    $startup
else
    xrb $OPENWINDHOME/lib/Xdefaults

    if [ -f $resources ]; then
        xrb -merge $resources
    fi

    xset -fp fonts/100dpi/
```

Listing 1 (Continued) /usr/xsrv/X11/lib/xdm/Xsession

```

xmodmap -e 'keysym F1 = Help'
xmodmap -e 'keysym F2 = F20'
xmodmap -e 'keysym F3 = F18'
xmodmap -e 'keysym F4 = F16'
xmodmap -e 'keysym F5 = F19'
xmodmap -e 'keysym F6 = F14'
xmodmap -e 'keysym F7 = F12'
xmodmap -e 'keysym F8 = F15'
xmodmap -e 'keysym F9 = F17'

o1wm 1
sleep 2

if ! -x $HOME/.openwin-init ; then
    SHOME/.openwin-init          # home startup
else
    $OPENWINHOME/lib/openwin-init # Default OpenWindows tools
fi

wait
fi

```

Listing 2 /usr/openwin/lib/openwin-menu

```

#
# @(#)openwin-menu      23.15 91/09/14 openwin-menu
#
#      OpenWindows default root menu file - top level menu
#

*Workspace* TITLE

*Programs* MENU      $OPENWINHOME/lib/openwin-menu-programs

*Utilities* MENU    $OPENWINHOME/lib/openwin-menu-utilities

*Properties...*     PROPERTIES

SEPARATOR

*Exit...*           EXIT

```

Listing 3 .openwin-menu

```

#
# @(#)openwin-menu
#
#      Private Workspace menu including own program menu.
#

```

Listing 3 (Continued) .openwin-menu

```

*Workspace* TITLE

*Programs* MENU      $OPENWINHOME/lib/openwin-menu-programs
*My Programs* MENU  $HOME/.openwin-menu-programs

*Utilities* MENU    $OPENWINHOME/lib/openwin-menu-utilities

*Properties...*     PROPERTIES

SEPARATOR

*Exit...*           EXIT

```

Listing 4 /usr/openwin/lib/openwin-menu-programs

```

#
# @(#)openwin-menu-programs      1.15 91/09/14 openwin-menu-programs
#
#      OpenWindows default root menu file - Programs submenu
#

*Programs* TITLE PIN

*xterm...*          exec $OPENWINHOME/bin/xterm
*Command Tool...*  DEFAULT exec $OPENWINHOME/bin/cmdtool
*vt340...*         exec /usr/xsrv/X11/bin/xtcoemx340
*t4397...*         exec /usr/xsrv/X11/bin/xtcoemx07
*Shell Tool...*    exec $OPENWINHOME/bin/shelltool

SEPARATOR

*Text Editor...*   exec $OPENWINHOME/bin/textedit
*File Manager...* exec $OPENWINHOME/bin/filemgr
*Mail Tool...*     exec $OPENWINHOME/bin/mailtool
*Calendar Manager...* exec $OPENWINHOME/bin/cm
*Clock...*         exec $OPENWINHOME/bin/clock
*Calculator...*    exec $OPENWINHOME/bin/calctool
*dbxtool...*       exec /usr/local/bin/xvview/dbxtool
*Print Tool...*    exec $OPENWINHOME/bin/printtool
*Tape Tool...*     exec $OPENWINHOME/bin/tapctool
*Snapshot...*      exec $OPENWINHOME/bin/snapshot
*Binder...*        exec $OPENWINHOME/bin/binder
*Icon Editor...*   exec $OPENWINHOME/bin/iconedit
*Performance Meter...* exec $OPENWINHOME/bin/perfimeter

*Demos* MENU      $OPENWINHOME/lib/openwin-menu-demos

```

Listing 5 /usr/openwin/lib/openwin-menu-utilities

```
#
# @(#)openwin-menu-utilities 1.5 91/09/14 openwin-menu-utilities
#
# OpenWindows default root menu file - Utilities submenu
#

*Refresh*          DEFAULT REFRESH
*Security...*      exec /usr/local/bin/secure
*Reset Input*     exec kbd_mode -u
*Function Keys*   exec vrb6
*Window Controls* MENU
  *Open/Close*    DEFAULT OPEN_CLOSE_SELN
  *Full/Restore Size* FULL_RESTORE_SIZE_SELN
  *Back*          BACK_SELN
  *Quit*          QUIT_SELN
*Window Controls* END PIN
*Save Workspace*  SAVE_WORKSPACE
#
# Uncomment the following if interested in colormap compaction
#
#*Save Colors* MENU
#  *Save*         DEFAULT $OPENWINHOME/bin/cmap_compact save
#  *Discard*      $OPENWINHOME/bin/cmap_compact discard
#*Save Colors* END
#
*Lock Screen*     exec xlock -remote -mode blank
```

