

OpenWindows Reference Manual

Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.



SunSoft
A Sun Microsystems, Inc. Business

© 1994 Sun Microsystems, Inc. All rights reserved.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® and Berkeley 4.3 BSD systems, licensed from UNIX Systems Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party software, including font technology, in this product is protected by copyright and licensed from Sun's Suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

This product or the products described herein may be protected by one or more U.S., foreign patents, or pending applications.

TRADEMARKS

Sun, Sun Microsystems, the Sun Logo, SunSoft, Sun Microsystems Computer Corporation and Solaris, are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. UNIX is a registered trademark of Novell, Inc., in the United States and other countries; X/Open Company, Ltd., is the exclusive licensor of such trademark. OPEN LOOK® is a registered trademark of Novell, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCcenter, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a product of the Massachusetts Institute of Technology.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Portions © AT&T 1983-1990 and reproduced with permission from AT&T.

| NAME | Intro – introduction to the OpenWindows Reference Manual | | | | | | | | | | | | | | |
|--------------------------|--|-------------|----------------------------|-----------------|--|--------------------|--|--------------------------|------------------------------------|----------------------|--|-------------------|---|----------------------|--|
| OVERVIEW | <p>The Solaris computing environment can be functionally divided into two large sub-components: the operating system and the windowing system. This Reference Manual describes the commands and utility programs which make up OpenWindows 3.4, the windowing system included with Solaris 2.4. The Solaris operating system commands are described in the <i>SunOS Reference Manual</i>.</p> <p>The OpenWindows Reference Manual consists of all the OpenWindows man pages included with the SUNWaman software package, other than those from Section 3. There are 187 man pages in this manual, most of which belong to Section 1 and describe general-purpose User commands. There are a few man pages which describe System Administration commands (Section 1M), and 28 man pages from Section 6, Games & Demonstration Programs. The SVR4 classification system for on-line man pages differs slightly from the BSD system, and so a few commands that used to belong in Section 5 have been moved to Section 4, which describes file formats. Not included in this manual are 764 man pages from Section 3, pages that describe functions and libraries. They can be accessed on systems with the SUNWaman package by using the man(1) command. For information on how to add a package see pkgadd(1).</p> <p>Since OpenWindows is based on the X11 display system, many of these man pages come directly from those included with X11R5 and have been modified only to correct pathname changes. For example, the pathname for /usr/lib/X11/Xinitrc has been changed to /usr/openwin/lib/Xinitrc. Several man pages in this manual come from Adobe Systems; they too have been edited only to correct for pathnames changes. The original formatting has been modified in order to give all pages a consistent appearance but no wording changes have been made except where strictly necessary.</p> <p>In addition to man pages which describe the X server (which include the DPS pages from Adobe Systems), this manual includes man pages about the DeskSet utility programs such as Calendar Manager and Printtool, as well as pages about AnswerBook, the Classing Engine, and ToolTalk.</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Name</th> <th style="text-align: left;">Command Description</th> </tr> </thead> <tbody> <tr> <td>24to8(1)</td> <td>convert a 24- or 32-bit rasterfile to an 8-bit rasterfile suitable for X11</td> </tr> <tr> <td>ab_admin(1)</td> <td>administer AnswerBook card catalog database(s)</td> </tr> <tr> <td>ab_cardcatalog(4)</td> <td>directory of available AnswerBooks</td> </tr> <tr> <td>ab_library(4)</td> <td>collection of AnswerBooks browsed and searched as a unit</td> </tr> <tr> <td>accessx(1)</td> <td>graphical interface to the accessx keyboard enhancements.</td> </tr> <tr> <td>answerbook(1)</td> <td>browse, search, and view on-line documentation</td> </tr> </tbody> </table> | Name | Command Description | 24to8(1) | convert a 24- or 32-bit rasterfile to an 8-bit rasterfile suitable for X11 | ab_admin(1) | administer AnswerBook card catalog database(s) | ab_cardcatalog(4) | directory of available AnswerBooks | ab_library(4) | collection of AnswerBooks browsed and searched as a unit | accessx(1) | graphical interface to the accessx keyboard enhancements. | answerbook(1) | browse, search, and view on-line documentation |
| Name | Command Description | | | | | | | | | | | | | | |
| 24to8(1) | convert a 24- or 32-bit rasterfile to an 8-bit rasterfile suitable for X11 | | | | | | | | | | | | | | |
| ab_admin(1) | administer AnswerBook card catalog database(s) | | | | | | | | | | | | | | |
| ab_cardcatalog(4) | directory of available AnswerBooks | | | | | | | | | | | | | | |
| ab_library(4) | collection of AnswerBooks browsed and searched as a unit | | | | | | | | | | | | | | |
| accessx(1) | graphical interface to the accessx keyboard enhancements. | | | | | | | | | | | | | | |
| answerbook(1) | browse, search, and view on-line documentation | | | | | | | | | | | | | | |

| | |
|--------------------------------|---|
| appres(1) | list application resource database |
| atobm(1) | See bitmap(1) |
| audiocontrol(1) | audio control panel |
| audiotool(1) | audio play/record tool |
| bdftopcf(1) | convert font from Bitmap Distribution Format to Portable Compiled Format |
| bdftosnf(1) | BDF to SNF font compiler for X11 |
| binder(1) | modify deskset bindings database |
| bitmap(1) | bitmap editor and converter utilities for the X Window System |
| bldrgb(1) | simple interface to the rgb(1) utility |
| bmtoa(1) | See bitmap(1) |
| calctool(1) | a desktop calculator |
| ce_db_build(1) | build a Classing Engine database from an ASCII description of the database or generate an ASCII description file of a CE database |
| ce_db_merge(1) | merge new information into a Classing Engine database from an ASCII description of the new information |
| clock(1) | display the time in an icon or window |
| cm(1) | calendar manager, appointment and resource scheduling tool |
| cmmap_alloc(1) | allocate default colormaps for non-default visuals |
| cmmap_compact(1) | colormap configuration utility to reduce colormap flashing. |
| cmdtool(1) | run a shell (or other program) in an OpenWindows enhanced terminal window |
| cm_delete(1) | delete appointments from Calendar Manager database |
| cm_insert(1) | insert appointments into Calendar Manager database |
| cm_lookup(1) | look up appointments from Calendar Manager database |
| CoEd(6) | a ToolTalk-based shared editor |
| colorchooser(1) | change icon colors in deskset tools |
| constype(1) | print type of Sun console |
| convert_to_Xdefaults(1) | convert SunView1 defaults into equivalent Xdefaults |
| convert_to_xview(1) | convert a SunView1 source program to XView source |
| digestp(1) | See mailp(1) |
| docviewer(1) | OpenWindows viewer for AnswerBook on-line documentation |
| dps(7) | Display PostScript imaging for the X Window System |
| dpsexec(6) | Display PostScript Executive |
| dsdm(1) | Drop Site Database Manager |

| | |
|---------------------------|--|
| ds_server_init (1) | store a property in the X11 server to reduce colormap flashing within the DeskSet applications |
| editres (1) | a dynamic resource editor for X Toolkit applications |
| edit_demo (6) | ToolTalk client/server demonstration programs |
| filemgr (1) | a GUI-based file management application |
| filep (1) | See mailp (1) |
| filofaxp (1) | See mailp (1) |
| fish_props (6) | See realxfishdb (6) |
| fixframe (1) | convert FrameMaker 2.0 PostScript files to conform to the PostScript structuring conventions |
| fixinterleaf (1) | convert Interleaf PostScript files to conform to the PostScript structuring conventions |
| fmgc (1) | a program to garbage collect a users ~/ .fm directory hierarchy |
| format_floppy (1) | diskette formatting program used by FileManager |
| franklinp (1) | See mailp (1) |
| helpopen (1) | utility for programmatically controlling helpviewer |
| helpviewer (1) | OpenWindows viewer for on-line help handbooks |
| ico (6) | animate an icosahedron |
| ico2 (6) | animate an icosahedron or other polyhedron |
| iconedit (1) | create and edit images for OpenWindows icons, cursors, and panel items |
| imagetool (1) | Image viewer for OpenWindows |
| imake (1) | C preprocessor interface to the make utility |
| kbd_mode (1) | change the keyboard translation mode |
| keytable.map (4) | maps keyboard type and layout to keytable |
| listres (1) | list resources in widgets |
| locale_env (1) | locale_env - program for openwin-sys |
| mailp (1) | frontends to the mp PostScript pretty printer |
| mailprint (1) | filter to strip out mail message attachments |
| mailtool (1) | OpenWindows interface for the mail program |
| makebdf (1) | create bitmap files from scalable F3 or X11/NeWS font files |
| makedepend (1) | create dependencies in makefiles |
| makepsres (1) | Build PostScript resource database file. |
| maze (6) | an automated maze program |
| mkdirhier (1) | makes a directory hierarchy |
| mkfontdir (1) | create fonts.dir file from directory of font files |

| | |
|-----------------------------|---|
| mp (1) | PostScript pretty printer |
| muncher (6) | draw interesting patterns in an X window |
| navigator (1) | browse and search AnswerBook on-line documentation |
| newsp (1) | See mailp (1) |
| oclock (1) | display time of day |
| olbiff (6) | OLIT mailbox monitor |
| olitsampler (6) | OLIT widget demo program |
| olitable (6) | OLIT widget demo program |
| olmh (6) | OpenLook interface to the MH message handling system |
| olwm (1) | OpenLook window manager for OpenWindows |
| olwmslave (1) | helper program for olwm |
| openwin (1) | OpenWindows startup script |
| owplaces (1) | list client applications running on a display |
| pageview (1) | PostScript language previewer for OpenWindows |
| perfmeter (1) | display system performance values in multiple dials or strip charts |
| plaid (6) | paint some plaid-like patterns in an X window |
| printtool (1) | OpenWindows tool for printing files |
| props (1) | graphical interface to set OpenWindows properties |
| pswrap (1) | creates C procedures from segments of PostScript language code |
| puzzle (6) | 15-puzzle game for X |
| ras2ps (1) | converts a Sun RasterFile to a PostScript file |
| rash (1) | Sun Raster to PostScript translator |
| rasterfile (4) | Sun's file format for raster images |
| realxfishdb (6) | Display a fishtank on the root window of an X11 server. |
| redxblue (1) | swap red and blue for a 24 or 32 bit rasterfile |
| resize (1) | utility to set TERM and terminal settings to current window size |
| rgb (1) | build the color name database |
| rpc.cmsd (1) | calendar manager service daemon |
| rpc.ttdbserverd (1M) | See ttdbserverd (1M) |
| sessreg (1) | manage utmp/wtmp entries for non-init clients |
| shelltool (1) | run a shell (or other program) in an OpenWindows terminal window |
| showrgb (1) | display the color name database |
| showsnf (1) | showsnf - print contents of an SNF file |
| snapshot (1) | capture some or all of a screen image and save to a raster file |

| | |
|-----------------------------|---|
| spider (6) | play double deck solitaire |
| tapetool (1) | OpenWindows tool for reading files from tape or archiving files to tape |
| textedit (1) | XView-based text editor with mouse support |
| textedit_filters (1) | filters provided with textedit(1) |
| texteroids (6) | test your mousing skills on spinning text |
| timemanp (1) | See mailp (1) |
| timesysp (1) | See mailp (1) |
| toolwait (1) | control client program startup |
| ttce2xdr (1M) | convert ToolTalk Classing Engine type tables to XDR format |
| ttcp (1) | copy files in a ToolTalk-safe way |
| ttdbck (1M) | display, check, or repair ToolTalk databases |
| ttdbserverd (1M) | RPC-based ToolTalk database server |
| ttmv (1) | move or rename files in a ToolTalk-safe way |
| ttrm (1) | remove files or directories in a ToolTalk-safe way |
| ttrmdir (1) | See ttrm (1) |
| ttsample1 (6) | simple ToolTalk demo |
| ttsession (1) | the ToolTalk message server |
| ttnoop (6) | the ToolTalk message monitor demo |
| tttar (1) | (de)archive files and ToolTalk objects |
| tt_type_comp (1) | the ToolTalk otype and ptype compiler |
| twm (1) | Tab Window Manager for the X Window System |
| viewprint (1) | print AnswerBook documents using lp |
| viewres (1) | graphical class browser for Xt |
| vkbd (1) | virtual keyboard/soft function keys display manager |
| winsysck (1) | check which window system protocols are available |
| worm (6) | draw wiggly worms |
| X (7) | See X11 (7) |
| X11 (7) | a portable, network-transparent window system |
| xauth (1) | X authority file utility |
| xbiff (1) | mailbox flag for X |
| xcalc (1) | scientific calculator for X |
| xclipboard (1) | X clipboard client |
| xclock (1) | analog/digital clock for X |
| xcmsdb (1) | Xlib Screen Color Characterization Data utility |

| | |
|--------------------------|--|
| xcolor (1) | displays 256 colors in an X window. |
| xconsole (1) | monitor system console messages |
| xcutsel (1) | interchange between cut buffer and selection |
| xditview (1) | display ditroff DVI files |
| xdm (1) | X Display Manager with support for XDMCP |
| xdpr (1) | dump an X window directly to a printer |
| xdpyinfo (1) | display information utility for X |
| xedit (1) | simple text editor for X |
| xepsf (6) | display an Encapsulated PostScript file |
| xev (6) | print contents of X events |
| xeyes (6) | Eyes follow your pointer |
| xfd (1) | display all the characters in an X font |
| xfontsel (1) | point & click interface for selecting X11 font names |
| xgc (6) | X graphics demo |
| xhost (1) | server access control program for X |
| xinit (1) | X Window System initializer |
| xkill (1) | kill a client by its X resource |
| xload (1) | system load average display for X |
| xlock (1) | Locks the local X display until a password is entered |
| xlogo (1) | X Window System logo |
| xlsatoms (1) | list interned atoms defined on server |
| xlsclients (1) | list client applications running on a display |
| xlsfonts (1) | server font list displayer for X |
| xlswins (1) | server window list displayer for X |
| xmac (6) | display Apple MacPaint image files under X windows |
| xmag (1) | magnify parts of the screen |
| xmag_multivis (6) | magnify parts of the screen |
| xmakemap (1) | make a keyboard mapping to be used as input to xmodmap |
| xman (1) | Manual page display program for the X Window System |
| xmh (1) | send and read mail with an X interface to MH |
| xmkmf (1) | simple interface to the imake utility for generating X11 Makefiles |
| xmodmap (1) | utility for modifying keymaps in X |
| xpr (1) | print an X window dump |
| xprop (1) | property displayer for X |
| xrdb (1) | X server resource database utility |

| | |
|-----------------------|--|
| xrefresh (1) | refresh all or part of an X screen |
| xscope (6) | X Window Protocol Viewer |
| Xserver (1) | X Window System server |
| xset (1) | user preference utility for X |
| xsetroot (1) | root window parameter setting utility for X |
| xsol (6) | play solitaire |
| xstdcmap (1) | X standard colormap utility |
| Xsun (1) | Solaris server for X Version 11 |
| xterm (1) | terminal emulator for X |
| xview (7) | xview toolkit information |
| xv_get_sel (1) | copy the content of a selection to the standard output |
| xwd (1) | dump an image of an X window |
| xwininfo (1) | window information utility for X |
| xwud (1) | image displayer for X |

| | |
|--------------------|--|
| NAME | 24to8 – convert a 24- or 32-bit rasterfile to an 8 bit rasterfile suitable for X11 |
| SYNOPSIS | 24to8 [-v] [-q] [-large] [<i>inrasf</i> -] [<i>outrasf</i>] |
| DESCRIPTION | <p>24to8 takes as input a 24- or 32-bit Sun rasterfile and reduces the depth of the image, from truecolor to 8 bit colormapped index color. 24to8 uses Floyd/Steinberg dithering to achieve high quality images. By default, this filter assumes that the color cube in the static colormap is of the form 5x5x5 (5 reds, 5 greens, and 5 blues).</p> <p>If both filenames are missing, the source rasterfile is read from stdin and the output rasterfile is written to stdout. If there is only one filename, then it is interpreted as the input rasterfile. To have a named output rasterfile and still read the input rasterfile from stdin, use a dash (-) in place of the input filename.</p> <p>24to8 interprets the input rasterfile as a packed 24-bit BGR format or an unpacked 32-bit XBGR-format; the standard Sun ordering for the red, green, and blue channels. This is the opposite of older versions of this program, which used the RGB or XBGR format. Older image files with the RGB format should be passed through the redxblue(1) filter prior to dithering with 24to8.</p> |
| OPTIONS | <p>-v Verbose mode will print information as it processes the image. (The default is to be silent.)</p> <p>-q Query (prints list of options)</p> <p>-large Dithers into a static colormap with a large color cube of the form 5x9x5.</p> |
| SEE ALSO | redxblue(1) |

| | |
|--------------------|---|
| NAME | ab_admin – administer AnswerBook card catalog database(s) |
| SYNOPSIS | ab_admin -file <i>card-catalog-file</i> -match <i>answerbook-id[,version]</i> -add <i>answerbook-id[,version]</i> <i>answerbook-attribute-list</i> -modify <i>answerbook-id[,version]</i> <i>answerbook-attribute-list</i> -merge <i>card-catalog-file</i> -convert <i>bookinfo-file</i> -remove <i>answerbook-id[,version]</i> -verify <i>answerbook-id[,version]</i> -list -listpaths |
| DESCRIPTION | ab_admin provides a shell-level interface to the AnswerBook card catalog database mechanism. It allows administrators to add, delete, list, modify, and verify individual AnswerBook entries in a given card catalog file. It also supports the merging of entries from one card catalog file into another, and the conversion of old-style "bookinfo" files into card catalog entries. See ab_cardcatalog (4) for more information on card catalogs |
| OPTIONS | -file <i>card-catalog-file</i> Specifies the name of the card catalog file. All ab_admin operations except "-listpaths" require that the file be specified. -merge <i>card-catalog-file</i> Merge the entries from input card catalog file into the card catalog file being updated. -convert <i>bookinfo-file</i> Extract the information from the specified bookinfo file needed to create a card catalog entry for the AnswerBook, and add that entry to the card catalog file being updated. -add <i>answerbook-id[,version]</i> [<i>answerbook-attribute-list</i>] Construct a new card catalog entry for the AnswerBook using the information provided on the command line. -remove <i>answerbook-id[,version]</i> Delete the specified entry. -match <i>answerbook-id[,version]</i> Display the contents of the specified entry. -modify <i>answerbook-id[,version]</i> [<i>answerbook-attribute-list</i>] Modify the specified fields of the specified entry. -verify <i>answerbook-id[,version]</i> Verify the correctness of the specified entry. Verify that the corresponding |

AnswerBook exists, and is usable.

-list List all entries in the specified card catalog.

-listpaths

List the fully qualified pathnames of the card catalogs you have access to, one per line.

AnswerBook Attributes

The following AnswerBook attributes are used in the **-add** and **-modify** operations of **ab_admin**. See **ab_cardcatalog(4)** for more information.

title=answerbook-title

Title of this AnswerBook

tocpath=toc-directory-path

Full pathname of the directory containing this AnswerBook's book database files

pspath=postscript-directory-path

Full pathname of the directory containing this AnswerBook's PostScript files

indexpath=index-directory-path

Full pathname of the directory containing this AnswerBook's search index files

EXAMPLES

See the *Software and AnswerBook Packages Administration Guide* for examples of **ab_admin(1)** usage.

SEE ALSO

ab_cardcatalog(4), **answerbook(1)**, **navigator(1)**

| | | | | | | | | | | | | | |
|-------------------------------|---|-------------------------------|----------------------------------|-------------------------|----------------------------------|----------------|--------------------------------------|---------------|--|----------------|---|-----------------|---|
| NAME | ab_cardcatalog – directory of available AnswerBooks | | | | | | | | | | | | |
| SYNOPSIS | \$HOME/.ab_cardcatalog, \$AB_CARDCATALOG | | | | | | | | | | | | |
| DESCRIPTION | <p>AnswerBook card catalogs serve as directories for listing and locating AnswerBooks on the local system and on the network. The AnswerBook programs navigator(1), docviewer(1), and others depend on card catalogs to list and find all available AnswerBooks.</p> <p>A card catalog file contains entries for one or more AnswerBooks. Each entry provides information on the AnswerBook's title, id, version number, and the location of its component files.</p> <p>Use the ab_admin(1) utility to add, delete, modify, or query AnswerBook entries in a card catalog file. Card catalog files are not intended for hand-editing, though they are ASCII files.</p> <p>AnswerBook programs that use card catalogs typically consult several of them, if necessary, in order to locate the entry for a given AnswerBook. In particular:</p> <table border="0" style="margin-left: 40px;"> <tr> <td>\$HOME/.ab_cardcatalog</td> <td>entries for personal AnswerBooks</td> </tr> <tr> <td>\$AB_CARDCATALOG</td> <td>shared, network-wide AnswerBooks</td> </tr> </table> <p>When an AnswerBook package is installed on a machine, the installation software will create a separate card catalog file for that particular AnswerBook. If that AnswerBook is to be shared among users in a network, its card catalog entry should be added to the network-wide card catalog file so that it will be generally available. See ab_admin(1) or the <i>Software and AnswerBook Packages Administration Guide</i> for more information on sharing AnswerBooks on the network.</p> | \$HOME/.ab_cardcatalog | entries for personal AnswerBooks | \$AB_CARDCATALOG | shared, network-wide AnswerBooks | | | | | | | | |
| \$HOME/.ab_cardcatalog | entries for personal AnswerBooks | | | | | | | | | | | | |
| \$AB_CARDCATALOG | shared, network-wide AnswerBooks | | | | | | | | | | | | |
| FORMAT | <p>Each AnswerBook entry in a card catalog file contains the following fields:</p> <table border="0" style="margin-left: 40px;"> <tr> <td>title</td> <td>AnswerBook title</td> </tr> <tr> <td>id</td> <td>AnswerBook id</td> </tr> <tr> <td>version</td> <td>AnswerBook version number (optional)</td> </tr> <tr> <td>pspath</td> <td>Full path name of directory containing AnswerBook's PostScript files</td> </tr> <tr> <td>tocpath</td> <td>Full path name of directory containing AnswerBook's Table of Contents databases</td> </tr> <tr> <td>indxpath</td> <td>Full path name of directory containing search index files</td> </tr> </table> <p>The "version" field is optional and serves mainly to distinguish two AnswerBooks that have the same id but different content. All other fields are mandatory</p> <p>The format of an entry is a set of colon-separated "field name=value" pairs. Entries can span more than one line by terminating each line with a backslash character ('\'). Blank lines and comment lines (those beginning with '#') are ignored.</p> | title | AnswerBook title | id | AnswerBook id | version | AnswerBook version number (optional) | pspath | Full path name of directory containing AnswerBook's PostScript files | tocpath | Full path name of directory containing AnswerBook's Table of Contents databases | indxpath | Full path name of directory containing search index files |
| title | AnswerBook title | | | | | | | | | | | | |
| id | AnswerBook id | | | | | | | | | | | | |
| version | AnswerBook version number (optional) | | | | | | | | | | | | |
| pspath | Full path name of directory containing AnswerBook's PostScript files | | | | | | | | | | | | |
| tocpath | Full path name of directory containing AnswerBook's Table of Contents databases | | | | | | | | | | | | |
| indxpath | Full path name of directory containing search index files | | | | | | | | | | | | |

The first line of a card catalog file must be the string

```
#<Card Catalog> version 1
```

which is the "magic number" for card catalog files. See **binder(1)** for more information on magic numbers.

Here is a sample card catalog file:

```
#<Card Catalog> version 1
```

```
#
```

```
# Sample card catalog entry for My AnswerBook
```

```
#
```

```
title=Title of My AnswerBook: \
```

```
id=MyAB: \
```

```
pspath=/net/my_server/export/AnswerBooks/MyAB/ps: \
```

```
tocpath=/net/my_server/export/AnswerBooks/MyAB/toc: \
```

```
indexpath=/net/my_server/export/AnswerBooks/MyAB/index
```

```
#
```

```
# Sample card catalog entry for Your AnswerBook
```

```
#
```

```
title=Title of Your AnswerBook: \
```

```
id=YourAB: \
```

```
pspath=/net/your_server/export/AnswerBooks/YourAB/ps: \
```

```
tocpath=/net/your_server/export/AnswerBooks/YourAB/toc: \
```

```
indexpath=/net/your_server/export/AnswerBooks/YourAB/index
```

SEE ALSO **ab_admin(1)**, **answerbook(1)**, **docviewer(1)**, **navigator(1)**, **viewprint(1)**

NOTES

Pre-Solaris 2.2 AnswerBooks use the old-style "bookinfo" mechanism, which has been superseded by card catalogs. The bookinfo files for these AnswerBooks should be converted to card catalog files using **ab_admin(1)** in order to work in OpenWindows Version 3.2 or higher.

AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc.

| | |
|--------------------|---|
| NAME | ab_library – collection of AnswerBooks browsed and searched as a unit |
| SYNOPSIS | \$HOME/.ab_library |
| DESCRIPTION | <p>An AnswerBook library is a group of AnswerBooks that can be browsed and searched as a unit by navigator(1) (the AnswerBook Navigator). Also included in a library is the list of bookmarks referencing that library's AnswerBooks.</p> <p>The default library file is \$HOME/.ab_library</p> <p>navigator automatically creates a new library for the user if one does not already exist. Whenever the AnswerBook user adds or deletes an AnswerBook to the library via the navigator "Modify Library" pop-up window, navigator saves those changes to the library file so that they exist between AnswerBook sessions. Changes to the library's bookmark list are similarly saved.</p> <p>The AnswerBook library file is created and maintained solely by the AnswerBook Navigator. It is not intended to be hand-editable.</p> <p>navigator employs a library file locking mechanism to prevent data loss or corruption when multiple Navigator access the same library file simultaneously. The Navigator creates a lock file (ab_library.lock). If a lock file already exists, the Navigator informs the user that the file is locked, then asks whether the user wants to 1) quit, 2) access the library in read-only mode, or 3) override the lock. Locks should only be overridden when they are known to be out of date, i.e., the program that created the lock is no longer running. The Navigator removes the lock file it created upon exiting.</p> |
| FILES | <p>\$HOME/.ab_library Default AnswerBook library file.</p> <p>\$HOME/.ab_library.lock AnswerBook library advisory locking file.</p> |
| SEE ALSO | navigator(1) |
| NOTES | AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc. |

| | |
|---------------------|---|
| NAME | accessx – A graphical interface to the accessx keyboard enhancements. |
| SYNOPSIS | accessx [-o] [-i] [-a] |
| AVAILABILITY | The keyboard enhancements and corresponding graphical interface are available on any UNIX workstation running the OpenWindows version 3.4 or later. |
| DESCRIPTION | accessx is a graphical user interface to enhancements within the OpenWindows 3.4 server. These enhancements provide the keyboard with additional capabilities which are primarily designed to aid users with disabilities who are unable to use a keyboard or mouse in the usual fashion. |
| OPTIONS | <p>-o Each user may load and save personal default settings for the various keyboard enhancement features. By default, the accessx utility will read in the user's default settings and configure the system to those settings. However, in certain instances the user might wish to use the systems current settings instead of his/her own default settings. Specifying the -o option will instruct the accessx utility to ignore a user's default settings and use the systems current settings instead.</p> <p>-i When this option is specified, the accessx user interface initially appears in an iconified state.</p> <p>-a Users may pop up dialog boxes indicating the state of Sticky Keys and Mouse Keys via the main menu of the accessx utility's graphical interface. Some users, however, want these status dialog windows to appear and disappear automatically whenever they turn on and off Sticky Keys or Mouse Keys. Specifying the -a option instructs the accessx utility to automatically pop up and down the Sticky Keys status window whenever Sticky Keys is turned on or off, respectively, and to automatically pop up and down the Mouse Keys status window whenever Mouse Keys is turned on or off, respectively.</p> |
| ENVIRONMENT | The HOME environment variable is used to identify the directory in which to load and save a user's default settings. The settings are stored in X resource file format in the file \$HOME/.AccessX. The user may modify or add resources to this file directly, however all comments will be deleted when the user performs a save settings action. Note that the UIDPATH environment variable should be set to /usr/openwin/lib/app-defaults/accessx.uid before starting accessx . |
| FILES | <p>accessx.uid - Motif GUI description file</p> <p>AccessX - Application Resource File</p> |
| SEE ALSO | X11(7) |
| AUTHORS | <p>Earl Johnson Jordan M. Slott Enabling Technologies</p> |

Sun Microsystems Laboratories, Inc
Mountain View, CA

Mark Novak
Trace Research & Development Center
University of Wisconsin-Madison
Madison, WI

Will Walker
Digital Equipment Corporation
Maynard, MA

| | |
|--------------------|---|
| NAME | answerbook – browse, search, and view on-line documentation |
| SYNOPSIS | answerbook [<i>navigator-options</i>] |
| DESCRIPTION | <p>answerbook starts up navigator(1) (the AnswerBook Navigator window) on the OpenWindows desktop after performing several tasks to initialize the AnswerBook environment. These tasks are important to the proper functioning of the AnswerBook software, so navigator should always be invoked via answerbook, rather than invoked directly by the user.</p> <p>answerbook performs the following functions each time it runs:</p> <p>Verify OpenWindows environment Makes sure that \$OPENWINHOME is set, that the correct version of OpenWindows (version 3.3 or greater) is running, etc.</p> <p>Source answerbook_setup If there is an executable sh or ksh shell script called "answerbook_setup" in the user's search path, source it. Typically, this file would contain shell commands that enable access to shared AnswerBooks available on the network. See the "AnswerBook Administration" guide for more information on sharing AnswerBooks on a network.</p> <p>Find locally installed AnswerBooks If there are any AnswerBooks installed on the local machine, find them (using pkginfo(1)) and make them available to the current AnswerBook session.</p> <p>Verify AnswerBooks present Make sure there is at least one AnswerBook available to the user in the current AnswerBook session. If not, print an error message and exit.</p> <p>Create Personal Library For first-time AnswerBook users, create a personal AnswerBook library file (\$HOME/.ab_library) that includes all the currently accessible AnswerBooks. See ab_library(4) for more information.</p> <p>Start AnswerBook Navigator Invoke navigator(1), passing on any command line options with which it was invoked.</p> |
| OPTIONS | answerbook passes all command line options on to navigator (1). |
| ENVIRONMENT | <p>AB_CARDCATALOG answerbook makes locally installed AnswerBooks accessible by appending their card catalog file names to the this environment variable, which navigator uses in assembling its list of available AnswerBooks. See ab_cardcatalog(4) for more information.</p> <p>OPENWINHOME This environment variable must be set appropriately.</p> |

| | |
|--------------------|--|
| FILES | <p>\$HOME/.ab_cardcatalog Personal card catalog file.</p> <p>\$HOME/.ab_library Default AnswerBook library file. Created automatically by answerbook for first-time AnswerBook users.</p> <p>answerbook_setup Shell script containing AnswerBook setup commands, typically for enabling access to shared AnswerBooks on the network. If there is a file by this name in the user's search path, answerbook sources it.</p> <p>/tmp/ab_cardcatalog.* Temporary card catalog file occasionally generated by answerbook to facilitate access to locally installed AnswerBooks.</p> |
| DIAGNOSTICS | <p>If answerbook finds no AnswerBooks during the initialization process, it prints a message to the console and exits without starting up the AnswerBook Navigator.</p> |
| SEE ALSO | <p>ab_admin(1), docviewer(1), navigator(1), ab_cardcatalog(4), ab_library(4)</p> |
| NOTES | <p>Pre-Solaris 2.2 AnswerBooks come with answerbook scripts of their own that are customized for that particular AnswerBook. These scripts do not work under OpenWindows 3.2, and are obviated by the new answerbook script.</p> <p>AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc.</p> |

| | |
|--------------------|---|
| NAME | appres – list application resource database |
| SYNOPSIS | appres [[class [instance]] [-1] [<i>toolkitoptions...</i>] |
| DESCRIPTION | <p>The appres program prints the resources seen by an application (or subhierarchy of an application) with the specified <i>class</i> and <i>instance</i> names. It can be used to determine which resources a particular program will load. For example,</p> <pre style="margin-left: 40px;">% appres XTerm</pre> <p>will list the resources that any xterm(1) program will load. If no application class is specified, the class <i>-AppResTest-</i> is used.</p> <p>To match a particular instance name, specify an instance name explicitly after the class name, or use the normal Xt toolkit option. For example,</p> <pre style="margin-left: 40px;">% appres XTerm myxterm</pre> <p>or</p> <pre style="margin-left: 40px;">% appres XTerm -name myxterm</pre> <p>To list resources that match a subhierarchy of an application, specify hierarchical class and instance names. The number of class and instance components must be equal, and the instance name should not be specified with a toolkit option. For example,</p> <pre style="margin-left: 40px;">% appres Xman.TopLevelShell.Form xman.topBox.form</pre> <p>will list the resources of widgets of xman(1) topBox hierarchy. To list just the resources matching a specific level in the hierarchy, use the -1 option. For example,</p> <pre style="margin-left: 40px;">% appres XTerm.VT100 xterm.vt100 -1</pre> <p>will list the resources matching the xterm(1) vt100 widget.</p> |
| SEE ALSO | X11 (7) xrdb (1), listres (1) |
| COPYRIGHT | Copyright 1989, Massachusetts Institute of Technology. See X11 (7) for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium |

| | |
|---------------------------|--|
| NAME | audiocontrol – audio control panel |
| SYNOPSIS | audiocontrol [-d device] [<i>generic-tool-arguments</i>] |
| DESCRIPTION | <p>Audio Control is an OpenWindows DeskSet application that controls and configures the workstation audio device. By default, Audio Control operates on /dev/audio, though the AUDIODEV environment variable can be used to override this default. An alternate audio device name may be entered on the command line using the -d option.</p> <p>Operations performed by the Audio Control control panel affect all audio programs using the specified device; for instance, adjusting the <i>Play Volume</i> instantly changes the output gain, regardless of which program is playing audio data. Further, Audio Control detects audio state changes made by other programs, and updates its display accordingly, so that it stays synchronized with the current device configuration.</p> <p>In addition to command line initiation, Audio Control may be started via the ToolTalk Media Exchange Protocol. Audio Tool, for instance, invokes Audio Control for volume and audio port control.</p> <p>Audio Control demonstrates an important principle involved in the integration of audio in the desktop environment: by enabling global control of important audio characteristics, it is not necessary for every application to provide an interface for these parameters. For instance, since the audio output can be set from the control panel, it is not strictly necessary that output applications contain output port controls themselves. However, such applications may detect that the audio output port has been changed, and take appropriate action.</p> |
| Play Control Panel | <p>Output Port</p> <p>This selector redirects audio output. There is a selector for each available output port provided by your hardware configuration. Normally, the available output ports include <i>Speaker</i> and <i>Headphone</i>. Some systems also support <i>Line Out</i>. If a control is present but inactivated (greyed-out), this indicates that the corresponding port is supported, but is unavailable; this can happen, for instance, when the audio device detects that a headset has not been plugged into the headphone jack. Note that some devices allow multiple output ports to be enabled simultaneously.</p> <p>Play Volume</p> <p>This slider adjusts the output volume. Volume levels between 0 and 100 may be selected, where 0 represents infinite attenuation and 100 is maximum gain.</p> <p>Play Balance</p> <p>This slider adjusts the left/right balance. This control does not appear if the audio device does not support stereo balance control.</p> <p>Mute This button is used to enable and disable audio output muting. Note that muting output has no effect on audio input recording.</p> <p>Record...</p> <p>This button brings up the Record Control Panel.</p> |

Record Control Panel

Input Port

This control selects the audio input port. There is a selector for each available input port provided by your hardware configuration. On many systems, only the *Microphone* port is available, although some systems also support *Line In*. If a control is present but inactivated (greyed-out), this indicates that the corresponding port is supported, but is unavailable; this can happen, for instance, when the audio device detects that a microphone has not been plugged into the microphone input jack. In general, input ports are mutually exclusive; that is, audio signals on multiple inputs are not mixed.

Record Volume

This slider adjusts the recording volume level in the range 0 to 100.

Record Balance

This slider adjusts the left/right balance. This control does not appear if the audio device does not support stereo balance control.

Auto-Adjust

Clicking this button starts an automatic recording level adjustment procedure. The input data is scanned to determine its loudness and the record volume is adjusted to achieve an optimal signal level. Once the input signal appears to be properly calibrated for several seconds, the panel will reset to a normal operating state. All input audio data is discarded during the automatic adjustment procedure.

A display gauge indicates the overall input level. If you have connected a microphone to the audio input jack, then you may adjust the recording level for your speaking voice by pressing the *Auto-Adjust* button and then speaking normally into the microphone until the adjustment process is finished. If you have connected a CD-player or other audio source, you can adjust the *Monitor Volume* to hear the audio data that is being used to calibrate the recording level.

Monitor Volume

This slider adjusts the monitor gain level in the range 0 to 100. Monitor gain controls the amount of audio input signal that is fed through to the output port. For instance, if an audio source (such as a radio or CD-player) is connected directly to the input port, the input signal may be monitored through the selected output port by adjusting this slider. Note that there may be audible feedback (a high-pitched whine) if a microphone is connected to the workstation and the monitor volume is set greater than zero.

Audio Device Status Panel

Audio Control includes an audio status panel that shows the current state of the audio device. This panel can be useful for debugging audio applications and determining whether applications are locking out the audio device by holding it open. Selecting "Status..." from the panel menu or pressing the **PROPS** (L3) key brings up the status panel.

Update When the *Update* selector is set to *Status Change*, the audio device status is updated only when a **SIGPOLL** signal is delivered to **Audio Control** (see **audio(7)**). Because of this, the *Active* and *Samples* indicators are not necessarily

kept up-to-date. This mode is useful for application debugging in order to see exactly when audio device status changes are being reported. When the *Continuous* mode is selected, the status is continually updated.

SEE ALSO [audiotool\(1\)](#), [audio\(7\)](#)

| | |
|------------------------------|--|
| NAME | audiotool – audio play/record tool |
| SYNOPSIS | audiotool [-p] [-d device] [<i>generic-tool-arguments</i>] [<i>audio-file ...</i>] |
| DESCRIPTION | Audio Tool is an OpenWindows DeskSet application for recording, playing, and simple editing of audio data. |
| Sound Segment Display | <p>Audio Tool analyzes sound data to distinguish between sound segments and silence segments. Segments of sounds are displayed as boxes; segments of silence are displayed as lines connecting these boxes. This type of display can be useful for scanning and editing voice data, since pauses often indicate breaks between sentences or key phrases. Data may be selected, using the mouse, in a manner similar to text selections. Double-clicking on a sound or silence segment selects the entire segment. The Cut, Copy, and Paste operations also work on selected data.</p> <p>A triangular graphic indicator, called the <i>position pointer</i>, marks the current play position. When you select a segment of audio data, the position pointer is constrained to remain within the selected segment. The position pointer also denotes the point in a segment where Record and Paste operations will insert data.</p> |
| Control Panel | <p>Play/Stop</p> <p>Clicking this button starts playing audio data at the current position. Playing will stop automatically at the end of the data or at the end of the currently selected segment, whichever comes first. If the position pointer is at the end of the data (or selection) when Play is pressed, it jumps back to the beginning of the data before starting to play.</p> <p>While data is playing, this button becomes a Stop button. Audio output may be stopped and restarted at will.</p> <p>Rewind/Fast-Forward</p> <p>Clicking either of these buttons causes the position pointer to jump backward or forward to the next sound segment boundary. If you press and hold one of these buttons, playing will begin/proceed at an accelerated rate in the designated direction. Releasing the button returns the tool to its previous state.</p> <p>Record/Stop</p> <p>Clicking this button starts the recording of data from the audio input port. The recorded data is inserted at the position pointer's position. If audio data is currently selected, it is replaced by the new recording. While recording is in progress, this button becomes a Stop button.</p> |
| File Browser Panel | <p>Selecting Open..., Save As..., or Include... from the File menu will bring up the File Browser Panel. The Directory field displays the pathname of the current directory.</p> <p>The scrolling list contains a list of directories or audio files in the displayed directory. Double-click SELECT on the name of a directory to change to that directory. Double-click SELECT on the name of an audio file to perform the current operation (Load, Save, or</p> |

Include) on the selected file. You can also type file names or directories into the **Name** field.

When **Save As...** is selected from the File menu, the File Browser displays an additional choice item, which enables you to select the file format. The choices are **Uncompressed** (8-bit 8 kHz, μ -law), or **Compressed** (4-bit G.721 ADPCM). The compressed format takes up half the disk space but can take somewhat longer to load and save.

Volume Panels

Selecting **Play...** or **Record...** from the **Volume** menu will invoke the **Audio Control** application to display volume control panels for the audio device. Refer to the **audiocontrol(1)** manual page for further details.

OPTIONS

-p Play all files listed on the command line at startup.

SEE ALSO

audiocontrol(1), **audioconvert(1)**, **audioplay(1)**, **audiorecord(1)**

Solaris User's Guide

About Audio Tool in the Help Handbook available through the Help option on the Workspace menu.

| | |
|--------------------|--|
| NAME | bdf _{topcf} – convert font from Bitmap Distribution Format to Portable Compiled Format |
| SYNOPSIS | bdf_{topcf} [-option ...] font-file.bdf |
| DESCRIPTION | Bdf_{topcf} is the release 5 font compiler. Fonts in Portable Compiled Format can be read by any architecture, although the file is structured to allow one particular architecture to read them directly without reformatting. This allows fast reading on the appropriate machine, but the files are still portable (but read more slowly) on other machines. |
| OPTIONS | <p>-pn Sets the font glyph padding. Each glyph in the font will have each scanline padded in to a multiple of <i>n</i> bytes, where <i>n</i> is 1, 2, 4 or 8.</p> <p>-un Sets the font scanline unit. When the font bit order is different from the font byte order, the scanline unit <i>n</i> describes what unit of data (in bytes) are to be swapped; the unit <i>i</i> can be 1, 2 or 4 bytes.</p> <p>-m Sets the font bit order to MSB (most significant bit) first. Bits for each glyph will be placed in this order; i.e. the left most bit on the screen will be in the highest valued bit in each unit.</p> <p>-l Sets the font bit order to LSB (least significant bit) first. The left most bit on the screen will be in the lowest valued bit in each unit.</p> <p>-M Sets the font byte order to MSB first. All multi-byte data in the file (metrics, bitmaps and everything else) will be written most significant byte first.</p> <p>-L Sets the font byte order to LSB first. All multi-byte data in the file (metrics, bitmaps and everything else) will be written least significant byte first.</p> <p>-t When this option is specified, <i>bdf_{topcf}</i> will convert fonts into "terminal" fonts when possible. A terminal font has each glyph image padded to the same size; the X server can usually render these types of fonts more quickly.</p> <p>-i This option inhibits the normal computation of ink metrics. When a font has glyph images which do not fill the bitmap image (i.e. the "on" pixels don't extend to the edges of the metrics) <i>bdf_{topcf}</i> computes the actual ink metrics and places them in the .pcf file; the -t option inhibits this behaviour.</p> <p>-o <output-file-name> By default <i>bdf_{topcf}</i> writes the pcf file to standard output; this option gives the name of a file to be used instead.</p> |
| SEE ALSO | X11(7) |
| COPYRIGHT | Copyright 1991, Massachusetts Institute of Technology. See <i>X11(7)</i> for a full statement of rights and permissions. |
| AUTHOR | Keith Packard, MIT X Consortium |

| | |
|--------------------|--|
| NAME | bdf2snf – BDF to SNF font compiler for X11 |
| SYNOPSIS | bdf2snf [-p#] [-u#] [-m] [-l] [-M] [-L] [-w] [-W] [-t] [-i] [<i>bdf-file</i>] |
| DESCRIPTION | bdf2snf reads a Bitmap Distribution Format (BDF) font from the specified file (or from standard input if no file is specified) and writes an X11 server normal font (SNF) to standard output. |
| OPTIONS | -p# Force the glyph padding to a specific number. The legal values are 1, 2, 4, and 8. -u# Force the scanline unit padding to a specific number. The legal values are 1, 2, and 4. -m Force the bit order to most significant bit first. -l Force the bit order to least significant bit first. -M Force the byte order to most significant bit first. -L Force the byte order to least significant bit first. -w Print warnings if the character bitmaps have bits set to one outside of their defined widths. -W Print warnings for characters with an encoding of -1; the default is to silently ignore such characters. -t Expand glyphs in "terminal-emulator" fonts to fill the bounding box. -i Don't compute correct ink metrics for "terminal-emulator" fonts. |
| WARNING | SNF is an older font file format that is replaced by the Portable Compiled Format (PCF). PCF is the preferred format for font files and one can use bdf2pcf(1) to convert a BDF font file to a PCF font file. |
| SEE ALSO | bdf2pcf(1) |

| | |
|---------------------|--|
| NAME | binder – modify deskset bindings database |
| SYNOPSIS | binder [-user -system -network] |
| AVAILABILITY | This command is available with the OpenWindows environment. |
| DESCRIPTION | <p>binder is an OpenWindows XView tool that allows the user to bind applications, icons, colors, print methods, and open methods to files. A binding is a logical connection between file types and elements such as file types, applications to be invoked when a file is opened, print scripts, or icons that the File Manager, Print Tool, Mail Tool, and other DeskSet applications use to display and operate on files.</p> <p>The binder displays all the bindings stored in three different Classing Engine databases: a network database, a system database, and a private user database. These are the Classing Engine databases used by all applications in the DeskSet Environment to determine how to display, print, and open any file. You cannot modify a system or network binding unless you are root.</p> <p>The three Classing Engine databases are located in: \$OPENWINHOME/lib/cetables/cetables, /etc/cetables/cetables, and \$HOME/.cetables/cetables, respectively.</p> <p>By doing this, users may customize their working environment without affecting others. Applications that use the bindings database in OpenWindows are:</p> <p>filemgr(1) binder(1) imagetool(1) printtool(1) mailtool(1)</p> |
| OPTIONS | <p>-user Modify your private database bindings. This is the default mode.</p> <p>-system Modify the system database bindings. You must be root to start the binder with this option.</p> <p>-network Modify the network database bindings. You must have root access on the OpenWindows server workstation.</p> |
| FILES | <p>\$OPENWINHOME/lib/cetables/cetables /etc/cetables/cetables \$HOME/.cetables/cetables</p> |
| BUGS | Changing your bindings will not automatically update all those tools that read the bindings database. Each tool will have to be quit and restarted for the new bindings to take effect. |
| SEE ALSO | <p>filemgr(1), ce_db_build(1), ce_db_merge(1) OpenWindows user documentation "About Binder" in the Help Handbook available through the Help option on the Workspace menu.</p> |

| | |
|-----------------------------|--|
| NAME | bitmap, bmtoa, atobm – bitmap editor and converter utilities for the X Window System |
| SYNOPSIS | bitmap [<i>-options ...</i>] [<i>filename</i>] [<i>basename</i>] bmtoa [<i>-chars ...</i>] [<i>filename</i>] atobm [<i>-chars cc</i>] [<i>-name variable</i>] [<i>-xhot number</i>] [<i>-yhot number</i>] [<i>filename</i>] |
| DESCRIPTION | <p>The bitmap program is a rudimentary tool for creating or editing 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.</p> <p>The bmtoa and atobm filters convert bitmap files (FILE FORMAT) to and from ASCII strings. They are most commonly used to quickly print out bitmaps and to generate versions for including in text.</p> |
| COMMAND LINE OPTIONS | <p>The bitmap command supports the standard X Toolkit command line arguments (see X11(7)). The following additional arguments are supported as well.</p> <p>-size <i>WIDTHxHEIGHT</i> Specifies size of the grid in squares.</p> <p>-sw <i>dimension</i> Specifies the width of squares in pixels.</p> <p>-sh <i>dimension</i> Specifies the height of squares in pixels.</p> <p>-gt <i>dimension</i> Grid tolerance. If the square dimensions fall below the specified value, grid will be automatically turned off.</p> <p>-grid, +grid Turns on or off the grid lines.</p> <p>-axes, +axes Turns on or off the major axes.</p> <p>-dashed, +dashed Turns on or off dashing for the frame and grid lines.</p> <p>-stippled, +stippled Turns on or off stippling of highlighted squares.</p> <p>-proportional, +proportional Turns proportional mode on or off. If proportional mode is on, square width is equal to square height. If proportional mode is off, bitmap will use the smaller square dimension, if they were initially different.</p> <p>-dashes <i>filename</i> Specifies the bitmap to be used as a stipple for dashing.</p> <p>-stipple <i>filename</i> Specifies the bitmap to be used as a stipple for highlighting.</p> |

-hl *color*

Specifies the color used for highlighting.

-fr *color*

Specifies the color used for the frame and grid lines.

filename

Specifies the bitmap to be initially loaded into the program. If the file does not exist, **bitmap** will assume it is a new file.

basename

Specifies the basename to be used in the C code output file. If it is different than the basename in the working file, **bitmap** will change it when saving the file.

The **bmtoa** command accepts the following option:

-chars *cc*

This option specifies the pair of characters to use in the string version of the bitmap. The first character is used for 0 bits and the second character is used for 1 bits. The default is to use dashes (-) for 0's and number signs (#) for 1's.

The **atobm** command accepts the following options:

-chars *cc*

This option specifies the pair of characters to use when converting string bitmaps into arrays of numbers. The first character represents a 0 bit and the second character represents a 1 bit. The default is to use dashes (-) for 0's and number signs (#) for 1's.

-name *variable*

This option specifies the variable name to be used when writing out the bitmap file. The default is to use the basename of the *filename* command line argument or leave it blank if the standard input is read.

-xhot *number*

This option specifies the X coordinate of the hotspot. Only positive values are allowed. By default, no hotspot information is included.

-yhot *number*

This option specifies the Y coordinate of the hotspot. Only positive values are allowed. By default, no hotspot information is included.

USAGE

The **bitmap** command displays grid in which each square represents a single bit in the picture being edited. Actual size of the bitmap image, as it would appear normally and inverted, can be obtained by pressing **Meta-I** key. You are free to move the image popup out of the way to continue editing. Pressing the left mouse button in the popup window or **Meta-I** again will remove the real size bitmap image.

If the bitmap is to be used for defining a cursor, one of the squares in the images may be designated as the hot spot. This determines where the cursor is actually pointing. For cursors with sharp tips (such as arrows or fingers), this is usually at the end of the tip; for symmetric cursors (such as crosses or bullseyes), this is usually at the center.

Bitmaps are stored as small C code fragments suitable for including in applications. They provide an array of bits as well as symbolic constants giving the width, height, and hot spot (if specified) that may be used in creating cursors, icons, and tiles.

EDITING

To edit a bitmap image simply click on one of the buttons with drawing commands (**Point, Curve, Line, Rectangle**, etc.) and move the pointer into the bitmap grid window. Press one of the buttons on your mouse and the appropriate action will take place. You can either set, clear or invert the grid squares. Setting a grid square corresponds to setting a bit in the bitmap image to 1. Clearing a grid square corresponds to setting a bit in the bitmap image to 0. Inverting a grid square corresponds to changing a bit in the bitmap image from 0 to 1 or 1 to 0, depending what its previous state was. The default behavior of mouse buttons is as specified below.

| | |
|--------------|--------|
| MouseButton1 | Set |
| MouseButton2 | Invert |
| MouseButton3 | Clear |
| MouseButton4 | Clear |
| MouseButton5 | Clear |

This default behavior can be changed by setting the button function resources. An example is provided below.

```
bitmap*button1Function: Set
bitmap*button2Function: Clear
bitmap*button3Function: Invert
etc.
```

The button function applies to all drawing commands, including copying, moving and pasting, flood filling and setting the hot spot.

**DRAWING
COMMANDS**

Here is the list of drawing commands accessible through the buttons at the left side of the application's window. Some commands can be aborted by pressing A inside the bitmap window, allowing the user to select different guiding points where applicable.

Clear

This command clears all bits in the bitmap image. The grid squares will be set to the background color. Pressing C inside the bitmap window has the same effect.

Set This command sets all bits in the bitmap image. The grid squares will be set to the foreground color. Pressing S inside the bitmap window has the same effect.

Invert

This command inverts all bits in the bitmap image. The grid squares will be inverted appropriately. Pressing I inside the bitmap window has the same effect.

Mark

This command is used to mark an area of the grid by dragging out a rectangular shape in the highlighting color. Once the area is marked, it can be operated on by a

number of commands (see **Up**, **Down**, **Left**, **Right**, **Rotate**, **Flip**, **Cut**, etc.) Only one marked area can be present at any time. If you attempt to mark another area, the old mark will vanish. The same effect can be achieved by pressing **Shift-MouseButton1** and dragging out a rectangle in the grid window. Pressing **Shift-MouseButton2** will mark the entire grid area.

Unmark

This command will cause the marked area to vanish. The same effect can be achieved by pressing **Shift-MouseButton3**.

Copy

This command is used to copy an area of the grid from one location to another. If there is no marked grid area displayed, **Copy** behaves just like **Mark** described above. Once there is a marked grid area displayed in the highlighting color, this command has two alternative behaviors. If you click a mouse button inside the marked area, you will be able to drag the rectangle that represents the marked area to the desired location. After you release the mouse button, the area will be copied. If you click outside the marked area, **Copy** will assume that you wish to mark a different region of the bitmap image, thus it will behave like **Mark** again.

Move

This command is used to move an area of the grid from one location to another. Its behavior resembles the behavior of **Copy** command, except that the marked area will be moved instead of copied.

Flip Horizontally

This command will flip the bitmap image with respect to the horizontal axes. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing **F** inside the bitmap window has the same effect.

Up This command moves the bitmap image one pixel up. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing **UpArrow** inside the bitmap window has the same effect.

Flip Vertically

This command will flip the bitmap image with respect to the vertical axes. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing **V** inside the bitmap window has the same effect.

Left

This command moves the bitmap image one pixel to the left. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing **LeftArrow** inside the bitmap window has the same effect.

Fold

This command will fold the bitmap image so that the opposite corners become adjacent. This is useful when creating bitmap images for tiling. Pressing **F** inside the bitmap window has the same effect.

Right

This command moves the bitmap image one pixel to the right. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing

RightArrow inside the bitmap window has the same effect.

Rotate Left

This command rotates the bitmap image 90 degrees to the left (counter clockwise.) If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing L inside the bitmap window has the same effect.

Down

This command moves the bitmap image one pixel down. If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing DownArrow inside the bitmap window has the same effect.

Rotate Right

This command rotates the bitmap image 90 degrees to the right (clockwise.) If a marked area of the grid is highlighted, it will operate only inside the marked area. Pressing R inside the bitmap window has the same effect.

Point

This command will change the grid squares underneath the mouse pointer if a mouse button is being pressed down. If you drag the mouse button continuously, the line may not be continuous, depending on the speed of your system and frequency of mouse motion events.

Curve

This command will change the grid squares underneath the mouse pointer if a mouse button is being pressed down. If you drag the mouse button continuously, it will make sure that the line is continuous. If your system is slow or **bitmap** receives very few mouse motion events, it might behave quite strangely.

Line

This command will change the grid squares in a line between two squares. Once you press a mouse button in the grid window, **bitmap** will highlight the line from the square where the mouse button was initially pressed to the square where the mouse pointer is located. By releasing the mouse button you will cause the change to take effect, and the highlighted line will disappear.

Rectangle

This command will change the grid squares in a rectangle between two squares. Once you press a mouse button in the grid window, **bitmap** will highlight the rectangle from the square where the mouse button was initially pressed to the square where the mouse pointer is located. By releasing the mouse button you will cause the change to take effect, and the highlighted rectangle will disappear.

Filled Rectangle

This command is identical to **Rectangle**, except at the end the rectangle will be filled rather than outlined.

Circle

This command will change the grid squares in a circle between two squares. Once you press a mouse button in the grid window, **bitmap** will highlight the circle from the square where the mouse button was initially pressed to the square where the mouse pointer is located. By releasing the mouse button you will cause the change

to take effect, and the highlighted circle will disappear.

Filled Circle

This command is identical to **Circle**, except at the end the circle will be filled rather than outlined.

Flood Fill

This command will flood fill the connected area underneath the mouse pointer when you click on the desired square. Diagonally adjacent squares are not considered to be connected.

Set Hot Spot

This command designates one square in the grid as the hot spot if this bitmap image is to be used for defining a cursor. Pressing a mouse button in the desired square will cause a diamond shape to be displayed.

Clear Hot Spot

This command removes any designated hot spot from the bitmap image.

Undo

This command will undo the last executed command. It has depth one, that is, pressing **Undo** after **Undo** will undo itself.

FILE MENU

The File menu commands can be accessed by pressing the File button and selecting the appropriate menu entry, or by pressing Ctrl key with another key. These commands deal with files and global bitmap parameters, such as size, basename, filename etc.

New

This command will clear the editing area and prompt for the name of the new file to be edited. It will not load in the new file.

Load

This command is used to load a new bitmap file into the bitmap editor. If the current image has not been saved, user will be asked whether to save or ignore the changes. The editor can edit only one file at a time. If you need interactive editing, run a number of editors and use cut and paste mechanism as described below.

Insert

This command is used to insert a bitmap file into the image being currently edited. After being prompted for the filename, click inside the grid window and drag the outlined rectangle to the location where you want to insert the new file.

Save

This command will save the bitmap image. It will not prompt for the filename unless it is said to be <none>. If you leave the filename undesignated or -, the output will be piped to stdout.

Save As

This command will save the bitmap image after prompting for a new filename. It should be used if you want to change the filename.

Resize

This command is used to resize the editing area to the new number of pixels. The

size should be entered in the WIDTHxHEIGHT format. The information in the image being edited will not be lost unless the new size is smaller than the current image size. The editor was not designed to edit huge files.

Rescale

This command is used to rescale the editing area to the new width and height. The size should be entered in the WIDTHxHEIGHT format. It will not do antialiasing and information will be lost if you rescale to the smaller sizes. Feel free to add your own algorithms for better rescaling.

Filename

This command is used to change the filename without changing the basename nor saving the file. If you specify - for a filename, the output will be piped to stdout.

Basename

This command is used to change the basename, if a different one from the specified filename is desired.

Quit

This command will terminate the bitmap application. If the file was not saved, user will be prompted and asked whether to save the image or not. This command is preferred over killing the process.

EDIT MENU

The Edit menu commands can be accessed by pressing the Edit button and selecting the appropriate menu entry, or by pressing Meta key with another key. These commands deal with editing facilities such as grid, axes, zooming, cut and paste, etc.

Image

This command will display the image being edited and its inverse in its actual size in a separate window. The window can be moved away to continue with editing. Pressing the left mouse button in the image window will cause it to disappear from the screen.

Grid

This command controls the grid in the editing area. If the grid spacing is below the value specified by gridTolerance resource (8 by default), the grid will be automatically turned off. It can be enforced by explicitly activating this command.

Dashed

This command controls the stipple for drawing the grid lines. The stipple specified by dashes resource can be turned on or off by activating this command.

Axes

This command controls the highlighting of the main axes of the image being edited. The actual lines are not part of the image. They are provided to aid user when constructing symmetrical images, or whenever having the main axes highlighted helps your editing.

Stippled

This command controls the stippling of the highlighted areas of the bitmap image. The stipple specified by stipple resource can be turned on or off by activating this command.

Proportional

This command controls the proportional mode. If the proportional mode is on, width and height of all image squares are forced to be equal, regardless of the proportions of the bitmap window.

Zoom

This command controls the zoom mode. If there is a marked area of the image already displayed, bitmap will automatically zoom into it. Otherwise, user will have to highlight an area to be edited in the zoom mode and bitmap will automatically switch into it. One can use all the editing commands and other utilities in the zoom mode. When you zoom out, undo command will undo the whole zoom session.

Cut This commands cuts the contents of the highlighted image area into the internal cut and paste buffer.

Copy

This command copies the contents of the highlighted image area into the internal cut and paste buffer.

Paste

This command will check if there are any other bitmap applications with a highlighted image area, or if there is something in the internal cut and paste buffer and copy it to the image. To place the copied image, click in the editing window and drag the outlined image to the position where you want to place it, and then release the button.

CUT AND PASTE

Bitmap supports two cut and paste mechanisms; the internal cut and paste and the global X selection cut and paste. The internal cut and paste is used when executing copy and move drawing commands and also cut and copy commands from the edit menu. The global X selection cut and paste is used whenever there is a highlighted area of a bitmap image displayed anywhere on the screen. To copy a part of image from another bitmap editor simply highlight the desired area by using the Mark command or pressing the shift key and dragging the area with the left mouse button. When the selected area becomes highlighted, any other applications (such as xterm, etc.) that use primary selection will discard their selection values and unhighlight the appropriate information. Now, use the Paste command for the Edit menu or control mouse button to copy the selected part of image into another (or the same) bitmap application. If you attempt to do this without a visible highlighted image area, the bitmap will fall back to the internal cut and paste buffer and paste whatever was there stored at the moment.

WIDGETS

Below is the widget structure of the **bitmap** application. Indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name. All widgets except the bitmap widget are from the standard Athena widget set.

```

    Bitmap bitmap
        TransientShell image
            Box box
                Label normalImage

```

```

        Label invertedImage
TransientShell input
    Dialog dialog
        Command okay
        Command cancel
TransientShell error
    Dialog dialog
        Command abort
        Command retry
TransientShell qsave
    Dialog dialog
        Command yes
        Command no
        Command cancel
Paned parent
    Form formy
        MenuButton fileButton
        SimpleMenu fileMenu
            SmeBSB new
            SmeBSB load
            SmeBSB insert
            SmeBSB save
            SmeBSB saveAs
            SmeBSB resize
            SmeBSB rescale
            SmeBSB filename
            SmeBSB basename
            SmeLine line
            SmeBSB quit
        MenuButton editButton
        SimpleMenu editMenu
            SmeBSB image
            SmeBSB grid
            SmeBSB dashed
            SmeBSB axes
            SmeBSB stippled
            SmeBSB proportional
            SmeBSB zoom
            SmeLine line
            SmeBSB cut
            SmeBSB copy
            SmeBSB paste
        Label status
    Pane pane
        Bitmap bitmap

```

Form form

Command clear
 Command set
 Command invert
 Toggle mark
 Command unmark
 Toggle copy
 Toggle move
 Command flipHoriz
 Command up
 Command flipVert
 Command left
 Command fold
 Command right
 Command rotateLeft
 Command down
 Command rotateRight
 Toggle point
 Toggle curve
 Toggle line
 Toggle rectangle
 Toggle filledRectangle
 Toggle circle
 Toggle filledCircle
 Toggle floodFill
 Toggle setHotSpot
 Command clearHotSpot
 Command undo

COLORS

If you would like bitmap to be viewable in color, include the following in the #ifdef COLOR section of the file you read with xrdb:

```
*customization:          -color
```

This will cause bitmap to pick up the colors in the app-defaults color customization file: /usr/openwin/lib/app-defaults/Bitmap.

BITMAP WIDGET

Bitmap widget is a stand-alone widget for editing raster images. It is not designed to edit large images, although it may be used in that purpose as well. It can be freely incorporated with other applications and used as a standard editing tool. The following are the resources provided by the bitmap widget.

Bitmap Widget

Header file Bitmap.h
 Class bitmapWidgetClass
 Class Name Bitmap
 Superclass Bitmap

All the Simple Widget resources plus ...

| Name | Class | Type | Default Value |
|-----------------|-----------------|-----------------|---------------------|
| foreground | Foreground | Pixel | XtDefaultForeground |
| highlight | Highlight | Pixel | XtDefaultForeground |
| framing | Framing | Pixel | XtDefaultForeground |
| gridTolerance | GridTolerance | Dimension | 8 |
| size | Size | String | 32x32 |
| dashed | Dashed | Boolean | True |
| grid | Grid | Boolean | True |
| stippled | Stippled | Boolean | True |
| proportional | Proportional | Boolean | True |
| axes | Axes | Boolean | False |
| squareWidth | SquareWidth | Dimension | 16 |
| squareHeight | SquareHeight | Dimension | 16 |
| margin | Margin | Dimension | 16 |
| xHot | XHot | Position | NotSet (-1) |
| yHot | YHot | Position | NotSet (-1) |
| button1Function | Button1Function | DrawingFunction | Set |
| button2Function | Button2Function | DrawingFunction | Invert |
| button3Function | Button3Function | DrawingFunction | Clear |
| button4Function | Button4Function | DrawingFunction | Invert |
| button5Function | Button5Function | DrawingFunction | Invert |
| filename | Filename | String | None ("") |
| basename | Basename | String | None ("") |

AUTHOR Davor Matic, X Consortium

NAME bldrgb – simple interface to the **rgb(1)** utility

SYNOPSIS **bldrgb**

DESCRIPTION **bldrgb** is a simple interface to the **rgb** utility which is used to build a color name database. **bldrgb** searches for the *rgb.txt* color name input file and then executes the **rgb** command, passing it the appropriate arguments.

EXAMPLES **example% bldrgb**

FILES

| | |
|----------------------------------|---|
| \$OPENWINHOME/lib/rgb.txt | color name database source. Maps color names to RGB color values. |
| \$OPENWINHOME/lib/rgb.dir | dbm file containing color name to RGB mapping. |
| \$OPENWINHOME/lib/rgb.pag | dbm file containing the color name to RGB mapping. |

SEE ALSO **cat(1), rgb(1), showrgb(1), dbm(3)**

| | |
|--------------------|---|
| NAME | calctool – a desktop calculator |
| SYNOPSIS | calctool [-2] [-3] [-a accuracy] [-c] [-l] [-m] [-name app-name] [-r] [-v] [-?] [-Wn] [+Wn] |
| DESCRIPTION | <p>calctool is a desktop calculator. It has been designed to be used with either the mouse or the keyboard. It is visually similar to a lot of hand-held calculators. There are financial, logical and scientific modes. Similar operations are color coded on color workstations. Some of the calculator keys have menu marks. This indicates that there is a menu associated with that key. Each key is discussed in more detail below.</p> <p>One of the most important things to remember about calctool is that calculations are performed from left to right, with no arithmetic precedence. If you need arithmetic precedence, then you should use parentheses (see below).</p> <p>Internal arithmetic is now done with multi-precision floating point numbers. Accuracy can be adjusted from zero to nine numeric places in fixed notation, but numbers can be displayed in engineering and scientific notation as well. The calculator reverts to scientific notation when the number is larger than the display would allow in fixed notation. The base of operation can be changed between binary, octal, decimal and hexadecimal. Numbers are initially displayed in fixed notation to two numeric places, in the decimal base.</p> <p>You can use the <i>Copy</i> and <i>Paste</i> functions in conjunction with the numeric display to store or retrieve characters from the text shelf. You can also remove the last digit entered, and the whole display can be cleared.</p> <p>There are ten memory registers. Numbers can be stored or retrieved in these locations, and arithmetic can be performed upon register contents.</p> <p>The display windows contains the current numerical value plus the current base and trigonometric type. There are also indicators which show if the hyperbolic and inverse function switches are set, and which numerical mode is currently in operation. If an operation needing more than one numerical input is partially complete, the operation is also displayed in this window as a reminder.</p> <p>On startup, calctool will use several X resources. These are listed in detail in the resources section of these manual pages.</p> <p>If you press MENU in the calctool window, and you aren't over a calculator key which has a menu associated with it, then a floating menu will appear, from which you can bring up a property sheet. Hitting the <i>Props</i> function key has the same effect.</p> <p>From this property sheet, you can modify several properties of the calculator. Each calculator button can have either a 2D or a 3D look, the keys can be in either monochrome or color (assuming you are using a color screen), and the buttons can be set to give either a "left-handed" or a "right-handed" appearance.</p> <p>Context sensitive help is also available. Position the mouse cursor over the item you need help with, and press the <i>Help</i> key. A popup will be displayed giving detailed information on this facility.</p> |

On startup, **calctool** will look for a **.calctoolrc** file in the users' home directory. This file allows the user to define personal constant and function definitions, plus setup the initial values for the ten memory registers. It then looks for a **.calctoolrc** file in the current directory, and if present, uses the contents of this file in place of any previously defined values.

With the **.calctoolrc** file, there are currently four valid record types. These are comments, constant and function definitions and initial memory register values.

Lines starting with a '#' are treated as comments and ignored.

Lines starting with 'c' or 'C' in the first column are definitions for constants. The cC is followed by a digit in the range 0-9, then a space. This is followed by a number in fixed or scientific notation. Following this is an optional comment, which if found, will be used in the popup menu for the constants. If the comment is present, then there must be at least one space between this and the preceding number.

Lines starting with 'f' or 'F' in the first column are definitions for functions. The fF is followed by a digit in the range 0-9, then a space. This is followed by a function definition. Following this is an optional comment, which if found, will be used in the popup menu for the functions. If the comment is present, then there must be at least one space between this and the preceding function definition.

Lines starting with 'r' or 'R' in the first column are definitions for the initial contents of the memory registers. The rR is followed by a digit in the range 0-9, then a space. This is followed by a number in fixed or scientific notation. The rest of the line is ignored.

All other lines are ignored. There should be no embedded spaces in the function definitions. Whenever a backslash is found, this and the following character signify a control character, for example \g would be ASCII 7.

OPTIONS

- 2 On a color screen, start **calctool** with a 2D look. This is the default.
- 3 On a color screen, start **calctool** with a 3D look.
- a *accuracy*
 Initial number of digits displayed after the numeric point. This value must be in the range 0 to 9. The default is two numeric places.
- c Display in color, assuming this is a color screen.
- l Start up a "left-handed" version of the **calctool** program.
- m Always display in monochrome, even on a color screen.
- name *app-name*
 This option specifies the application name under which resources are to be obtained, rather than the default executable file name. *app-name* should not contain "." or "*" characters.
- r Start up a "right-handed" version of the **calctool** program. This is the default.
- v Show the version number and the usage message of this release of the **calctool** program.
- ? Show the version number and the usage message of this release of the **calctool**

program. Note that the ? character must be escaped if using **cs**h(1).

-Wn Start **calctool** with no title line.

+Wn Start **calctool** with a title line present.

RESOURCES

On startup, **calctool** uses the following resources:

Resource: deskset.calctool.accuracy

Values: Accuracy value (numeric)

Description The number of of digits displayed after the numeric point. This value must be in the range 0 to 9.

Resource: deskset.calctool.base

Values: Numeric Base (string)

Description The initial numeric base. Valid values are binary, octal, decimal and hexadecimal.

Resource: deskset.calctool.display

Values: Display mode (string)

Description The initial display mode. Valid values are engineering, fixed-point and scientific.

Resource: deskset.calctool.mode

Values: Mode (string)

Description The initial calculator mode. Valid values are basic, financial, logical and scientific.

Resource: deskset.calctool.mono

Values: True, False (False)

Description Whether the calculator should start in monochrome. Color is obviously only allowed on screens that support it.

Resource: deskset.calctool.beep

Values: True, False (True)

Description Indicates whether a beep should sound in the event of an error (such as invalid input, or if the display needs to be cleared).

Resource: deskset.calctool.showRegisters

Values: True, False (True)

Description Whether the memory register window is initially displayed.

Resource: deskset.calctool.trigType

Values: Trig. type (string)

Description The initial trigonometric type. Valid values are degrees, grads and radians.

| | |
|--------------------|--|
| Resource: | deskset.calctool.rightHanded |
| Values: | True, False (True) |
| Description | Whether the calculator is started with a "right-handed" display style. |
| Resource: | deskset.calctool.3dLook |
| Values: | True, False (True) |
| Description | If the display is color, whether each calculator button has a pseudo 3D appearance. |
| Resource: | deskset.calctool.hasTitle |
| Values: | True, False (True) |
| Description | Indicates whether the calctool window has a title line. |
| Resource: | deskset.calctool.buttonFont |
| Values: | Font name string |
| Description | The name of the font used to display all button labels, irrespective of the current size and scale of calctool . |
| Resource: | deskset.calctool.modeFont |
| Values: | Font name string |
| Description | The name of the font used to display the various mode labels, irrespective of the current size and scale of calctool . |
| Resource: | deskset.calctool.memoryFont |
| Values: | Font name string |
| Description | The name of the font used to display the memory register values, irrespective of the current size and scale of calctool . |
| Resource: | deskset.calctool.displayFont |
| Values: | Font name string |
| Description | The name of the font used to show the current display value, irrespective of the current size and scale of calctool . |
| Resource: | deskset.calctool.decDigitColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the buttons containing the decimal digits 0 - 9 and the numeric point on the main panel. |
| Resource: | deskset.calctool.hexDigitColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the buttons containing the hexadecimal digits A - F on the main panel. |
| Resource: | deskset.calctool.arithOpColor |
| Values: | Color name string or hexadecimal color specification string |

| | |
|--------------------|---|
| Description | The color of the arithmetic operator buttons on the main panel. |
| Resource: | deskset.calctool.adjustColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the Bsp and Clr buttons on the main panel. |
| Resource: | deskset.calctool.portionColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the Abs, Chs, Frac and Int buttons on the main panel. |
| Resource: | deskset.calctool.functionColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of various function buttons on the main panel. These are Acc, Con, Exch, Fun, Keys, Mem, Quit, Rcl and Sto. |
| Resource: | deskset.calctool.mainModeColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the three mode buttons on the main panel. These are Base, Disp and Mode. |
| Resource: | deskset.calctool.portionLogicalColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the logical buttons in the mode panel that do shift operations or take a portion of the result. These are <, >, &16 and &32. |
| Resource: | deskset.calctool.bitLogicalColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the bitwise logical operator buttons in the mode panel. These are And, Not, Or, Xnor and Xor. |
| Resource: | deskset.calctool.finColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the financial buttons in the mode panel. |
| Resource: | deskset.calctool.trigModeColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the trigonometrical mode buttons in the mode panel. These are Hyp, Inv and Trig. |
| Resource: | deskset.calctool.trigColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the trigonometrical buttons in the mode panel. These are Cos, Sin and Tan. |
| Resource: | deskset.calctool.sciColor |

| | |
|---------------------|---|
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the scientific buttons in the mode panel. These are e^x , 10^x , y^x , $x!$, Ln, Log and Rand. |
| Resource: | deskset.calctool.backgroundColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the background area for the calculator buttons in the main and mode panels. |
| Resource: | deskset.calctool.displayColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the numerical display area in the main panel. |
| Resource: | deskset.calctool.memRegisterColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The background color in the memory register window. |
| Resource: | deskset.calctool.textColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of all text. This includes the numerical display, the button labels and the contents of the memory register window. |

CALCULATOR BUTTONS

This section describes the calculator keys present in the main **calctool** window. Apart from this basic mode, **calctool** has three other modes, and the keys associated each each of these modes are described in separate sections below.

Keyboard equivalents appear in the square brackets. Note that '^' followed by a letter indicates that the Control key and this key should be pressed together.

Numerical Keys [0-9 a-f . = <Return>].

Enter a digit (decimal digits 0-9 or hexadecimal digits A-F) into the display. The '.' character acts as the numeric point, and '=' (or Return) is used to complete numerical entry. Upto forty digits may be entered.

Arithmetical Operations [+ - x * /].

Perform an arithmetical operation using the previous entry and the next entry as operands. Addition, subtraction, multiplication and division are denoted by the characters '+', '-', 'x' and '/' respectively ('*' is also synonymous with multiplication).

Number Manipulation Operators.

| | | |
|-------------|--------|---|
| Int | [^i] | Return the integer portion of the current entry. |
| Frac | [^f] | Return the fractional portion of the current entry. |
| Abs | [^u] | Return the absolute value of the current entry. |
| +/- | [C] | Change the arithmetic sign of the current entry. |
| 1/x | [r] | Return the value of 1 divided by the current entry. |
| x^2 | [@] | Return the square of the current entry. |
| % | [%] | Perform a percentage calculation using the last entry and the next entry. |
| Sqrt | [s] | Perform a square root operation on the current entry. |

Menu Operations.

Asc [^a] If Asc is selected with the mouse, then a separate window is displayed which allows you to enter any character. The ASCII value of this character is then displayed in the current base. If this option is selected via the keyboard, then you don't get a special window displayed.

Each of these operations has a popup menu associated with it. This menu can be displayed using the MENU mouse button, and a selection made. You can select the default item from the menu using the SELECT mouse button.

It is also possible to use just the keyboard to achieve the same results. The first keyboard value selects the menu operation; the second keyboard character selects the new value for this operation. Unlike the menu facility available with the mouse, there is no visual feedback on what choices are available to you, so the user has to know what item they wish to select.

Base [B] Change the base that calculations are displayed in. The available choices are binary [b], octal [o], decimal [d] and hexadecimal [h]. Digits that are inappropriate for a particular base selection are greyed out in the main **calctool** window.

Disp [D] Change the numerical display mode. The choices are engineering [e], fixed point [f], and scientific [s] notation.

Mode [M] Change the calculator mode. By default, **calctool** is in basic mode, and just the keys on the main **calctool** window are visible. There are also three other modes; financial [f], logical [l] and scientific [s]. Selection of one of these modes will display an extra window with more keys. These special operations are described in more detail in the sections below.

Acc [A] Set the display accuracy. Between 0 and 9 [0-9] significant digits can be displayed.

Con [#] Retrieve and display a constant value. There are ten constant values [0-9], and each one has a default value which can be overridden by entries in the users **.calctoolrc** file. The ten default values are:

| | | |
|---|-----------|--------------------------------|
| 0 | 0.621 | kms per hour / miles per hour. |
| 1 | 1.41421 | square root of 2. |
| 2 | 2.71828 | e. |
| 3 | 3.14159 | pi. |
| 4 | 2.54 | cms / inch. |
| 5 | 57.29578 | degrees in a radian. |
| 6 | 1048576.0 | 2 to the power of 20. |
| 7 | 0.0353 | gms / oz. |
| 8 | 0.948 | kilojoules / British thermals. |
| 9 | 0.0610 | cubic cms / cubic inches. |

Fun [F] Retrieve and execute a function expression. There are ten function definitions [0-9]. These are setup with entries in the users **.calctoolrc** file.

Other Operations.

| | | |
|----------------|----------------|--|
| Rcl | [R] | Retrieve memory register value. There are ten memory registers [0-9]. |
| Sto | [S] | Store value in memory register. There are ten memory registers [0-9]. The register number may be preceded by an arithmetic operation (addition, subtraction, multiplication or division), in which case the specified operation is carried out between the displayed entry and the value currently in the selected memory register, and the result is placed in the memory register. |
| Exch | [X] | Exchange the current display with the contents of a memory register. There are ten memory registers [0-9]. |
| Clr | [Delete] | Clear the calculator display. |
| Bsp | [Back Space] | Remove the rightmost character of the current entry, and recalculate the displayed value. |
| (and) | [(and)] | Parentheses. Allow precedence with arithmetic calculations. Note that parentheses can be nested to any level, and calctool provides a visual feedback of what is being typed in, in the calculator display. The calculation doesn't take place until the last parenthesis is matched, then the display is updated with the new result. |
| Exp | [E] | This is used to allow numbers to be entered in scientific notation. The mantissa should be initially entered, then the Exp key selected. The exponent is then entered. If no numerical input had occurred when the Exp key was selected, then a mantissa of 1.0 is assumed. |
| Keys | [k] | Toggle the labels on the calctool buttons between the mouse and keyboard equivalents. |
| Mem | [m] | Display the window with the ten memory register values. These values are displayed in the current base to the current degree of accuracy using the current numerical display notation. |
| Quit | [q or Q] | Exit without user verification. |

FINANCIAL
MODE

An example of how to use each of these financial calculations, is available via the context sensitive help facility.

| | | |
|-------------|--------|---|
| Ctrm | [^t] | Compounding term. Computes the number of compounding periods it will take an investment of present value pv to grow to a future value of fv, earning a fixed interest rate int per compounding period. Memory register usage: Register 0 int (periodic interest rate). Register 1 fv (future value). Register 2 pv (present value). |
| Ddb | [^d] | Double-declining depreciation. Computes the depreciation allowance on an asset for a specified period of time, using the double-declining balance method. Memory register usage: Register 0 cost (amount paid for asset). Register 1 salvage (value of asset at end of life). |

| | | | | |
|-------------|--------|--|---------|---|
| | | Register 2 | life | (useful life of the asset). |
| | | Register 3 | period | (time period for depreciation allowance). |
| Fv | [v] | Future value. This calculation determines the future value of an investment. It computes the future value based on a series of equal payments, each of amount pmt, earning periodic interest rate int, over the number of payment periods in term. | | |
| | | Memory register usage: | | |
| | | Register 0 | pmt | (periodic payment). |
| | | Register 1 | int | (periodic interest rate). |
| | | Register 2 | n | (number of periods). |
| Pmt | [P] | Periodic payment. Computes the amount of the periodic payment of a loan. Most installment loans are computed like ordinary annuities, in that payments are made at the end of each payment period. | | |
| | | Memory register usage: | | |
| | | Register 0 | prin | (principal). |
| | | Register 1 | int | (periodic interest rate). |
| | | Register 2 | n | (term). |
| Pv | [p] | Present value. Determines the present value of an investment. It computes the present value based on a series of equal payments, each of amount pmt, discounted at periodic interest rate int, over the number of periods in term. | | |
| | | Memory register usage: | | |
| | | Register 0 | pmt | (periodic payment). |
| | | Register 1 | int | (periodic interest rate). |
| | | Register 2 | n | (term). |
| Rate | [^r] | Periodic interest rate. Returns the periodic interest necessary for a present value of pv to grow to a future value of fv over the number of compounding periods in term. | | |
| | | Memory register usage: | | |
| | | Register 0 | fv | (future value). |
| | | Register 1 | pv | (present value). |
| | | Register 2 | n | (term). |
| Sln | [^s] | Straight-line depreciation. Computes the straight-line depreciation of an asset for one period. The straight-line method of depreciation divides the depreciable cost (cost - salvage) evenly over the useful life of an asset. The useful life is the number of periods (typically years) over which an asset is depreciated. | | |
| | | Memory register usage: | | |
| | | Register 0 | cost | (cost of the asset). |
| | | Register 1 | salvage | (salvage value of the asset). |
| | | Register 2 | life | (useful life of the asset). |
| Syd | [^y] | Sum-of-the-years-digits depreciation. The sum-of-the-years'-digits method of depreciation accelerates the rate of depreciation, so that | | |

more depreciation expense occurs in earlier periods than in later ones. The depreciable cost is the actual cost minus salvage value. The useful life is the number of periods (typically years) over which an asset is depreciated.

Memory register usage:

Register 0 cost (cost of the asset).

Register 1 salvage (salvage value of the asset).

Register 2 life (useful life of the asset).

Register 3 period (period for which depreciation is computed).

Term [T] Payment period. Returns the number of payment periods in the term of an ordinary annuity necessary to accumulate a future value of fv, earning a periodic interest rate of int. Each payment is equal to amount pmt.

Memory register usage:

Register 0 pmt (periodic payment).

Register 1 fv (future value).

Register 2 int (periodic interest rate).

LOGICAL MODE

< [<] Shift the current entry to the left. The shift can be between 1 and 15 places [1-9, a-f]. This calculator key has a popup menu associated with it.

> [>] Shift the current entry to the right. The shift can be between 1 and 15 places [1-9, a-f]. This calculator key has a popup menu associated with it.

&16 [] Truncate the current entry to a 16 bit unsigned integer.

&32 [[] Truncate the current entry to a 32 bit unsigned integer.

Or [|] Perform a logical OR operation on the current entry and the next entry, treating both numbers as unsigned long integers.

And [&] Perform a logical AND operation on the current entry and the next entry, treating both numbers as unsigned long integers.

Not [~] Perform a logical NOT operation on the current entry.

Xor [^] Perform a logical XOR operation on the current entry and the next entry, treating both numbers as unsigned long integers.

Xnor [n] Perform a logical XNOR operation on the current entry and the next entry, treating both numbers as unsigned long integers.

SCIENTIFIC MODE

Trig [T] Set the current trigonometrical base. This can be in degrees [d], grads [g] or radians [r]. This key also has a popup menu associated with it.

Hyp [h] Toggle the hyperbolic function indicator. This switch affects the type of sine, cosine and tangent trigonometric functions performed.

Inv [i] Toggle the inverse function indicator. This switch affects the type of sine, cosine and tangent trigonometric functions performed.

e^x [{] Returns e raised to the power of the current entry.

| | | |
|-----------------------|--------|--|
| 10^x | []] | Returns 10 raised to the power of the current entry. |
| y^x | [y] | Take the last entry and raise it to the power of the next entry. |
| x! | [!] | Return the factorial of the current entry. Note that the factorial function is only valid for positive integers. |
| Cos | [^c] | Return the trigonometric cosine, arc cosine, hyperbolic cosine or inverse hyperbolic cosine of the current display, depending upon the current settings of the hyperbolic and inverse function switches. The result is displayed in the current trigonometric units (degrees, radians or grads). |
| Sin | [^s] | Return the trigonometric sine, arc sine, hyperbolic sine or inverse hyperbolic sine of the current display, depending upon the current settings of the hyperbolic and inverse function switches. The result is displayed in the current trigonometric units (degrees, radians or grads). |
| Tan | [^t] | Return the trigonometric tangent, arc tangent, hyperbolic tangent or inverse hyperbolic tangent of the current display, depending upon the current settings of the hyperbolic and inverse function switches. The result is displayed in the current trigonometric units (degrees, radians or grads). |
| Ln | [N] | Return the natural logarithm of the current entry. |
| Log | [G] | Return the base 10 logarithm of the current entry. |
| Rand | [?] | Return a random number between 0.0 and 1.0. |

FILES

`~/desksetdefaults` stored X resources.
`.calctoolrc` user's personal calctool startup file.

SEE ALSO

Solaris User's Guide
 "About Calculator" in the Help Handbook available through the **Help** option on the Workspace menu.

| | |
|--------------------|---|
| NAME | ce_db_build – build a Classing Engine database from an ASCII description of the database or generate an ASCII description file of a CE database |
| SYNOPSIS | ce_db_build <i>user</i> <i>system</i> <i>network</i> -from_ascii -to_ascii <i>filename</i> [-db_file <i>db_filename</i>] |
| DESCRIPTION | <p>ce_db_build builds a Classing Engine database from an ASCII description file <i>or</i> generate an ASCII description file of a CE database. The Classing Engine implements a simple hierarchy of databases. Each database contains tables of namespaces. Each namespace table allows for namespace specific mappings of names to attributes. For example, the Files namespace allows for mappings from file names to file attributes; e.g., file type.</p> <p>An example of the contents of an ASCII description file is:</p> <pre> { NS_NAME=Types NS_ATTR=((NS_MANAGER,string,<\$CEPATH/tns_mgr.so>)) NS_ENTRIES=(((TYPE_NAME,type-id,<binder-prog>) (TYPE_ICON,icon-file,<\$OPENWINHOME/include/images/bind.icon>) (TYPE_ICON_MASK,icon-file,<\$OPENWINHOME/include/images/bind.mask.icon>) (TYPE_FGCOLOR,color,<91 229 229>) (TYPE_BGCOLOR,color,<91 126 229>) (TYPE_PRINT,string,<lp -Plp>))) (TYPE_NAME,type-id,<calctool-prog>) (TYPE_ICON,icon-file,<\$OPENWINHOME/include/images/calctool.icon>) (TYPE_ICON_MASK,icon-file,<\$OPENWINHOME/include/images/calctool.mask.icon>) (TYPE_FGCOLOR,color,<255 255 255>) (TYPE_BGCOLOR,color,<229 45 183>) (TYPE_PRINT,string,<lp -Plp>))) } { NS_NAME=Files NS_ATTR=((NS_MANAGER,junk,<\$CEPATH/fns_mgr.so>)) NS_ENTRIES=(((FNS_TYPE,refto-Types,<binder-prog>) (FNS_FILENAME,str,<binder>)))) </pre> |

```
(FNS_TYPE,ref-to-Types,<default-app>)
(FNS_FILENAME,str,<default|app>)
)
)
}
```

In the above example:

- * The ASCII description file defines two namespaces called Types and Files.
- * The keyword NS_NAME precedes the name of a namespace.
- * The keyword NS_ATTR precedes a list of the attributes of the namespace.

An important namespace attribute is called NS_MANAGER, which identifies the name of the shared library (see **dlopen(3X)**) that implements the namespace manager for the namespace.

NS_ENTRIES precedes a list of entries in the namespace. Each entry consists of a list of attributes.

An attribute has a name, a type and a value.

Attribute names in the example are in upper case e.g. TYPE_NAME, FNS_TYPE. Attribute types are just strings that make sense to the CE client. For example, FNS_TYPE has a type of "ref-to-Types" - an indication to the CE client that FNS_TYPE values are really TYPE_NAMES that can be looked up in Types namespace.

A BNF for the ASCII description file is:

```
database      : name_space | database name_space
name_space    : { name namespace_attributes entries }
name          : NS_NAME = variable
ns_attrs      : NS_ATTR = ( attribute_list )
manager_attribute : ( NS_MANAGER , attribute_type , attribute_value )
attribute_list : attribute
                | attribute_list attribute
attribute      : ( attribute_name , attribute_type , attribute_value )
entries        : NS_ENTRIES = ( list_of_entries )
list_of_entries : an_entry
                | list_of_entries an_entry
an_entry       : ( attribute_list )
attribute_name : variable
attribute_type : variable
variable       : Identifier
attribute_value : attribute_token
```

The terminals are NS_NAME, NS_ATTR, NS_MANAGER, NS_ENTRIES, "{", "}", "(", ")", " ", "=", Identifier and attribute_token. Identifier can have a-z, A-Z, 0-9, _, -.

attribute_token can come in two flavors:

It can begin with a "<" and end with a ">" and can have any ASCII character (except a ">") within it.

It can begin with one or more digits (which represent a number "n") followed by zero or more spaces, followed by a "<", followed by any "n" characters, closed off by a ">". This is the escape mechanism to allow for arbitrary byte string attributes that could have ">" characters within them.

OPTIONS

user | system | network

Indicates whether the user wants to access the user, system or network CE database.

-from_ascii filename

Indicates that the user wishes to write the stated CE database from the ASCII file *filename*. The entire CE database will be re-written; i.e. this is an all or nothing update of the CE database.

-to_ascii filename

Indicates that the file named *filename* should be written with the ASCII description of the stated CE database. This ASCII description can be modified or added to and supplied as input to an invocation of **ce_db_build** with the **-from_ascii** argument.

db_file db_filename

This form should be used in the case that a particular database is to be read from/written to using *db_filename* as the pathname of the CE database.

ENVIRONMENT

CEPATH

This is a colon separated list of up to three pathnames that the CE uses when looking for a CE database to read or write. It also uses CEPATH as a list of places to look if a particular NS_MANAGER filename has CEPATH prepended to it.

The first pathname is for the "user" database, the second pathname is for the "system" database, and the third pathname is for the "network" database.

If a CEPATH is not defined, or if a particular pathname is not specified in CEPATH, the following database pathnames are used by default:

If the database is:

"user" — ~/.cetables/cetables

"system" — /etc/cetables/cetables

"network" — \$OPENWINHOME/lib/cetables/cetables

For example, to set the pathname for the "system" database to /foo/cetables while using the defaults for "user" and "network", set CEPATH to:

:/foo/cetables

FILES

cetables

This is the Classing Engine database file produced in the appropriate directory after **ce_db_build** is complete. A CE database file can be identified by its first

characters which state:
"Classing Engine Data File Version 1.0aaa"

SEE ALSO `ce_db_merge(1)`

BUGS If the CE database file you are building is mounted from a pre-SunOS 4.1 machine, the locking protocol used by `ce_db_build` does not work; i.e., the database does not get locked for writing.

Running `ce_db_build` on an empty ASCII file causes it to hang indefinitely as though it were in an infinite loop.

| | |
|--------------------|---|
| NAME | ce_db_merge – merge new information into a Classing Engine database from an ASCII description of the new information |
| SYNOPSIS | ce_db_merge <i>user</i> <i>system</i> <i>network</i> -from_ascii <i>filename</i> [-db_file <i>db_filename</i>] |
| DESCRIPTION | <p>ce_db_merge merges new information into a Classing Engine database from an ASCII description file. The Classing Engine implements a simple hierarchy of databases. Each database contains tables of namespaces. Each namespace table allows for namespace specific mappings of names to attributes. For example, the Files namespace allows for mappings from file names to file attributes; e.g., file type.</p> <p>The ASCII description for merge has the same syntax as that used for building a database from scratch (see ce_db_build(1)).</p> <p>An example ASCII file is:</p> <pre> { NS_NAME=Types NS_ATTR=((NS_MANAGER,string,<\$CEPATH/tns_mgr.so>)) NS_ENTRIES=(((TYPE_NAME,type-id,<binder-prog>) (TYPE_ICON,icon-file,<\$OPENWINHOME/include/images/bind.icon>) (TYPE_ICON_MASK,icon-file,<\$OPENWINHOME/include/images/bind.mask.icon>) (TYPE_FGCOLOR,color,<91 229 229>) (TYPE_BGCOLOR,color,<91 126 229>) (TYPE_PRINT,string,<lp -Plp>))) (TYPE_NAME,type-id,<calctool-prog>) (TYPE_ICON,icon-file,<\$OPENWINHOME/include/images/calctool.icon>) (TYPE_ICON_MASK,icon-file,<\$OPENWINHOME/include/images/calctool.mask.icon>) (TYPE_FGCOLOR,color,<255 255 255>) (TYPE_BGCOLOR,color,<229 45 183>) (TYPE_PRINT,string,<lp -Plp>))) } { NS_NAME=Files NS_ATTR=((NS_MANAGER,junk,<\$CEPATH/fns_mgr.so>)) NS_ENTRIES=(((FNS_TYPE,ref-to-Types,<binder-prog>) (FNS_FILENAME,str,<binder>))) </pre> |

```

(FNS_TYPE,ref-to-Types,<default-app>)
(FNS_FILENAME,str,<default|app>)
)
)
}

```

In the above example:

- * The ASCII description file defines two namespaces called Types and Files.
- * The keyword NS_NAME precedes the name of a namespace.
- * The keyword NS_ATTR precedes a list of the attributes of the namespace.

An important namespace attribute is called NS_MANAGER, which identifies the name of the shared library (see **dlopen(3X)**) that implements the namespace manager for the namespace.

NS_ENTRIES precedes a list of entries in the namespace. Each entry consists of a list of attributes.

An attribute has a name, a type and a value.

Attribute names in the example are in upper case; e.g., TYPE_NAME, FNS_TYPE. Attribute types are just strings that make sense to the CE client. For example, FNS_TYPE has a type of "ref-to-Types" - an indication to the CE client that FNS_TYPE values are really TYPE_NAMES that can be looked up in Types namespace.

A BNF for the ASCII description file is:

```

database      : name_space | database name_space
name_space    : { name namespace_attributes entries }
name          : NS_NAME = variable
ns_attrs      : NS_ATTR = ( attribute_list )
manager_attribute : ( NS_MANAGER , attribute_type , attribute_value )
attribute_list : attribute
               | attribute_list attribute
attribute     : ( attribute_name , attribute_type , attribute_value )
entries       : NS_ENTRIES = ( list_of_entries )
list_of_entries : an_entry
               | list_of_entries an_entry
an_entry      : ( attribute_list )
attribute_name : variable
attribute_type : variable
variable      : Identifier
attribute_value : attribute_token

```

The terminals are NS_NAME, NS_ATTR, NS_MANAGER, NS_ENTRIES, "{", "}", "(", ")", " ", "=", Identifier and attribute_token. Identifier can have a-z, A-Z, 0-9, _, -.

attribute_token can come in two flavors:

It can begin with a "<" and end with a ">" and can have any ASCII character (except a ">") within it.

It can begin with one or more digits (which represent a number "n") followed by zero or more spaces, followed by a "<", followed by any "n" characters, closed off by a ">". This is the escape mechanism to allow for arbitrary byte string attributes that could have ">" characters within them.

If the database already had a namespace defined (e.g., Types), the new entries would just be appended to the existing namespace with no checking for duplicate entries.

If the namespace description has some namespace attributes defined (e.g., NS_MANAGER) that already exist in the database, the new namespace attributes overwrite the existing attributes.

OPTIONS

user | system | network

Indicates whether the user wants to access the user, system or network CE database.

-from_ascii filename

Indicates that the user wishes to merge the stated CE database from the ASCII file *filename*.

-db_file db_filename

This form should be used in the case that a particular database is to be read from/written to using *db_filename* as the pathname of the CE database.

ENVIRONMENT

CEPATH

This is a colon separated list of up to three pathnames that the CE uses when looking for a CE database to read or write. It also uses CEPATH as a list of places to look if a particular NS_MANAGER filename has CEPATH prepended to it.

The first pathname is for the "user" database, the second pathname is for the "system" database, and the third pathname is for the "network" database.

If a CEPATH is not defined, or if a particular pathname is not specified in CEPATH, the following database pathnames are used by default:

If the database is:

"user" — ~/.cetables/cetables

"system" — /etc/cetables/cetables

"network" — \$OPENWINHOME/lib/cetables/cetables

For example, to set the pathname for the "system" database to /foo/cetables while using the defaults for "user" and "network", set CEPATH to:

:/foo/cetables

FILES

cetables

This is the Classing Engine database file modified in the appropriate directory after **ce_db_merge** is complete. A CE database file can be identified by its first

characters which state:
"Classing Engine Data File Version 1.0aaa"

SEE ALSO `ce_db_build(1)`

NOTES `ce_db_merge` appends entries to the namespace. Merged entries will be masked by existing entries. To overcome an existing entry, use `ce_db_build -to_ascii`, edit the ascii file, then use `ce_db_build -from_ascii`.

An entry in the "network" database can be masked by an entry in the "system" database. Likewise, an entry in the "system" database can be masked by an entry in the "user" database.

BUGS If the CE database file you are merging is mounted from a pre-SunOS 4.1 machine, the locking protocol used by `ce_db_merge` does not work; i.e., the database does not get locked for writing.
Running `ce_db_merge` on an empty ASCII file causes it to hang indefinitely as though it were in an infinite loop.

| | |
|---------------------|--|
| NAME | clock – display the time in an icon or window |
| SYNOPSIS | clock [-Wn] [+Wn] [-T] [-TZ <i>timezone</i>] [-12] [-24] [-alarm <i>setting</i>] [-alarmtime <i>hr:min</i>] [-alarmcmd <i>cmd</i>] [-analog] [-digital] [+date] [-date] [-help] [-hourcmd <i>cmd</i>] [-name <i>app-name</i>] [-r] [+seconds] [-seconds] [-v] [-?] |
| AVAILABILITY | Although this command is available with both the SunView software installation option and the OpenWindows environment, its appearance is not the same for both environments. The man page you are currently viewing refers to the clock that is available with OpenWindows. The primary difference between this and the SunView clock is the user interface. All OpenWindows standard tools use the OPEN LOOK Graphical User Interface. |
| DESCRIPTION | clock is an OpenWindows XView utility that displays the current time in a window or icon. When the window is open, clock can display the time in either analog or digital format. The clock window is scalable in either format. |
| OPTIONS | <p>-Wn Start clock with no title line.</p> <p>+Wn Start clock with a title line present.</p> <p>-T Start the clock in test mode. Test mode ignores the real time, and instead runs in a loop continuously incrementing the time by one minute and displaying it.</p> <p>-TZ <i>timezone</i> Start the clock with an alternate time zone.</p> <p>-12 If the digital clock is being displayed in the open window, then use a 12 hour clock.</p> <p>-24 If the digital clock is being displayed in the open window, then use a 24 hour clock.</p> <p>-alarm <i>setting</i> Set how often you want the alarm to go off. Choices are "none", "once" and "daily".</p> <p>-alarmtime <i>hr:min</i> The time you want the alarm to go off.</p> <p>-alarmcmd <i>cmd</i> The command you want to run, when the alarm does go off.</p> <p>-analog Start up the open window display with the analog clock.</p> <p>-digital Start up the open window display with the digital clock.</p> <p>+date Show the current date in the open window.</p> <p>-date Do not show the current date in the open window.</p> |

- help** Lists generic XView options that can be applied to the **clock**.
- hourcmd** *cmd*
The command you would like run every hour, on the hour.
- name** *app-name*
This option specifies the application name under which resources are to be obtained, rather than the default executable file name. *app-name* should not contain "." or "*" characters.
- r** Use a face with roman numerals in the iconic state. This replaces the default round face.
- +seconds**
Show the number of seconds. For the analog clock (and in the iconic state), the seconds are shown by displaying a second hand. For the digital display, seconds are shown as a numeric value.
- seconds**
Do not show the number of seconds.
- v** Show the version number and the usage message for this version of **clock**.
- ?** Show the version number and the usage message for this version of **clock**.

USAGE

When the **clock** window is open, it has a floating menu from which you can bring up a property window. From this property sheet, you can modify the open window clock face of the tool by selecting either the **Analog** or **Digital** choice box.

The icon displayed, can be either an **analog** or **roman** clock face. With the digital display, you can select between a **12-hour** or a **24-hour** display.

Seconds:

On the analog version, this selection starts a second hand on the face of the clock. On the digital version, it adds two digits to the digital readout.

Date:

Turns on a date display for both analog and digital versions of **clock**.

You can select a local timezone or set it to monitor a remote timezone by clicking SELECT on other, then choosing a timezone by pressing MENU over the abbreviated menu button displayed.

The stopwatch is an automatic function, and allows you to reset, start and stop a stopwatch to second accuracy. This should be set to none, to disable it.

An alarm can be set, and when that time occurs, then a command can be run. The alarm can be triggered once or daily. If no alarm command is present, then the **clock** will just beep.

You can also set a command that will be run every hour on the hour.

There are various keyboard accelerators that can be used with the **clock**. These are:

- 1** set 12hr mode if currently showing a digital display.
- 2** set 24hr mode if currently showing a digital display.
- c** toggle clock face between analog and digital.

- d** toggle on and off the date display.
- i** toggle icon face between analog and roman.
- s** toggle on and off the seconds display.
- S** set stopwatch. Cycles between reset, start and stop. Hit c to continue in normal clock display.
- t** toggle the timezone between local and other. Note that you have to have a valid other timezone for this to work properly.
- T** toggle test mode on and off.
- q** quit the clock

On startup, the **clock** will use the following X resources:

- | | |
|--------------------|--|
| Resource: | deskset.clock.faceAnalog |
| Values: | True, False (True) |
| Description | Setting for the open clock face. Choices are analog and digital. |
| | |
| Resource: | deskset.clock.iconAnalog |
| Values: | True, False (True) |
| Description | Setting for the closed clock icon. Choices are analog and roman. |
| | |
| Resource: | deskset.clock.iconHasWindowColor |
| Values: | True, False (False) |
| Description | Determines whether the clock icon should use the window background color. By default, it uses the workspace color. |
| | |
| Resource: | deskset.clock.digital12Hour |
| Values: | True, False (True) |
| Description | Setting for digital displays. Choices are 12 hour and 24 hour display. |
| | |
| Resource: | deskset.clock.showLocal |
| Values: | True, False (True) |
| Description | Indicates whether the local or other timezone should be initially displayed. |
| | |
| Resource: | deskset.clock.secondHand |
| Values: | True, False (False) |
| Description | When True, a second hand (for analog) or a seconds value (for digital) will be displayed. |
| | |
| Resource: | deskset.clock.date |
| Values: | True, False (False) |
| Description | When True, the current date will be displayed. |
| | |
| Resource: | deskset.clock.timeZone |
| Values: | Timezone (string) |

| | |
|------------------|---|
| Description | If present, the timezone to monitor. |
| Resource: | deskset.clock.alarmHrValue |
| Values: | Hour value (numeric) |
| Description | The hour value for the alarm command. |
| Resource: | deskset.clock.alarmMinValue |
| Values: | Minute value (numeric) |
| Description | The minute value for the alarm command. |
| Resource: | deskset.clock.alarmChoice |
| Values: | None, Once, Daily (None) |
| Description | How often the alarm should go off. Choices are none, once and daily. |
| Resource: | deskset.clock.alarmCommand |
| Values: | Command (string) |
| Description | The command that should be run when the alarm goes off. |
| Resource: | deskset.clock.hourlyCommand |
| Values: | Command (string) |
| Description | The command that should be run every hour, on the hour. |
| Resource: | deskset.clock.dateFont |
| Values: | Font name string |
| Description | The name of the font used to display the date and timezone, irrespective of the current size and scale of clock |
| Resource: | deskset.clock.secondsFont |
| Values: | Font name string |
| Description | The name of the font used to display the number of seconds when in digital mode, plus the am/pm values if in 12 hour mode, irrespective of the current size and scale of clock |
| Resource: | deskset.clock.hasTitle |
| Values: | True, False (True) |
| Description | Indicates whether the clock window has a title line. |

ENVIRONMENT

The **clock** uses the **TZ** environment variable to determine the local time, and the initial setting for the other timezone. The latter can be over-ridden in a variety of ways.

SEE ALSO

OpenWindows user documentation
 "About the Clock" in the Help Handbook available through the Help option on the Workspace menu.

| | |
|---------------------|---|
| NAME | cm – calendar manager, appointment and resource scheduling tool |
| SYNOPSIS | cm [<i>generic-tool-arguments</i>] [-c <i>calendar</i>] [-i [2-3] |
| AVAILABILITY | The man page you are currently viewing refers to the cm that is available with OpenWindows. All OpenWindows standard tools use the OPEN LOOK Graphical User Interface (GUI). |
| DESCRIPTION | <p>cm is an appointment and resource scheduling tool that allows you to do the following:</p> <ul style="list-style-type: none"> - Display day, week, month, year views - Schedule single or repeating appointments and todo items - Browse and edit calendars of other users - Set beeping, flashing, or pop-up reminders - Integrate with mail - Restrict access to your own calendar - Print high-quality hardcopy - View and Schedule appointments on other calendars - Change the time zone context <p>A full appointment editor is available for inserting and editing appointments. Most settings on the appointment editor can be preloaded through a Properties window available from the main panel.</p> <p>Menu accelerators exist for some commonly used menu options: View:Find, Edit:Appointment, Edit:Properties, and Print:Current View.</p> <p>A holiday list can be loaded by creating a file with a name ending in .cal in the user's home directory containing lines of the following form:</p> <p>month/day holiday-name</p> |
| OPTIONS | <p>-c <i>calendar</i> The name of the default calendar. The calendar specified overrides the default user-name or the Default Calendar specified in the properties of Calendar Manager.</p> <p>generic-tool-arguments cm accepts the generic tool arguments described in xview(7).</p> <p>-i [2-3] You have the option of using the OWv2 or OWv3 icon. The OWv2 icon has the current calendar month displayed. The OWv3 icon has the current date displayed. The default is the OWv3 icon, so unless -i 2 is used, the OWv3 icon will be used.</p> |
| USAGE | <p>cm operates via a set of pulldown menus from button stacks in a control panel. The menu commands are described below.</p> <p>The Control Area</p> |

View Menu

- Day** Displays the Day View. Also displays the Appointments and Todo items in a window for the selected day.
- Week** Displays the Week View. Also displays the Appointments and Todo items in a window for the selected week. A chart is also displayed with the times for the appointments shaded.
- Month** Displays the Month View. Also displays the Appointments and Todo items in a window for the month.
- Year** Displays a year calendar and the appointments and Todo items for a year.
- Time Zone** Sets time zones. The current times are changed to reflect the new time zone.
- Find...** Searches for an appointment string. A time range can be specified for the search. It is case insensitive and will match any part of the appointment 'what:' fields.
- Go To...** Displays the calendar at the date requested.

Edit Menu**Appointment**

Schedule appointments by either selecting the day and choosing Appointment from the Edit menu or by double-clicking on day (in the month or week view) or on an hour (in the day view). Inserts, deletes and changes appointments.

Date: The date is automatically set by the tool when a day is selected. You may change the date format through the property settings.

Start: and Stop:

Sets the time of the appointment. A choice of times is found on the pulldown menus. The choices are determined by the Day Bounds settings from the Properties window. You may set the time format through the Property settings to be on a 12 or 24 hour clock.

What: Filled in with information about the appointment. This information will be displayed on the various views.

Alarm: Determines what action will be taken in advance of the appointment.

Beep Produces an audible sound.

Flash Inverts the frame of the tool or the icon if the tool is closed.

Popup Pops up a reminder window with the text of the appointment showing.

Mail Composes a message with the text of the appointment and sends it to recipients in the

Mail To field.

Appointment | Todo

Specifies whether the event is an appointment or todo item. Can later be displayed via the View menu.

DND Drop Target

Allows the user to drag appointments onto the appointment editor.

Appointments Scrolling List

Automatically set with the current appointments for that day. It will be updated as appointments are added and deleted.

Insert Access Scrolling List

Displays a list of people with insert access

Repeat:

Indicates whether the appointment is a repeating event of the specified interval. The number of times the appointment repeats is controlled by the For panel item.

Privacy:

Determines whether others who view the calendar will see both the time and text of the appointment, the time only, or nothing at all.

Insert Inserts the appointment into the calendar.

Delete Deletes the appointment selected in the scrolling list from the calendar.

Change

Changes the appointment selected in the scrolling list to the edited values from the appointment editor.

For Insert, Delete, and Change, a Forward feature has

been added that allows the user to delete only occurrences of the appointment in the future.

Reset Clears the appointment editor of context and sets default values from the Properties window.

Properties

Sets default values for this tool. These values are used for setting default values in the Appointment Editor, setting display characteristics of the views, granting access to the calendar, setting printer options and date formats.

Editor Defaults

Sets default reminder values and advance times for appointments. Also sets the default names for which mail will be sent

when a mail reminder is set. Sets the default privacy for appointments.

Display Settings

Sets default begin and end times for the day that are used in drawing the chart on the week view and the grid on the day view. The User calendar Location allows the user to set the location of the callog file. Sets a 12 or 24 hour clock. Sets the default view that will appear on start up. Sets the calendar to the Default Calendar specified on start up.

Access List and Permissions

Sets permissions for a list of remote calendars. The default is 'World Browse'. These entries will be available using the **Browse** pulldown menu after they are added to the list and the changes are applied. Access can be Browse (Read), Insert, and/or Delete. The keyword **world**, opens up the calendar to everyone. The entries have the format **name@host**. At this time, UNIX aliases are not supported.

Printer Settings

Sets the default printer settings The Privacy Type allows the user to decide the default Privacy Type to be printed.

Date Format

Sets the default date format including the ordering of the month, day and year and the date separator.

Apply Adds any changes made.

Reset Reestablishes the values before you started editing.

Defaults

Reestablishes the tool default values.

Browse Menu

The default menu item is the user name.

Show Multiple Calendars

Allows easy method for finding common open time slots among a group of users. Highlighting the names in the scrolling list will cause the time slots in each user's calendar to be highlighted if an appointment is scheduled at that time. The browser chart represents the degree of 'busy-ness' with the darker shades representing a larger number of conflicts. The Schedule button brings up the appointment editor. The Mail button brings up a mail compose window with the highlighted names in the To: field. It includes as an attachment an appointment icon which makes it easy to schedule an appointment. The Go To: button allows easy navigation through the calendars. The Setup Menu button brings up the Setup Menu popup.

Show Calendar

Brings up a popup which allows the user to type in the user name of a calendar to browse.

Setup Menu

Allows the user to add, change, remove, or sort names in the Browse Menu. Any changes made to this list are also made to the browse menu pulldown and the Multiple Calendars scrolling list after the user has pressed Apply.

Add Adds a name to the list.

Change Changes the selected item in the list to the value in the User Name: text field.

Remove Removes a name from the list.

Sort List Sorts the list.

Apply Permanently incorporates any changes made to this list, the browse menu pulldown, and the Multiple Calendars scrolling list.

Reset Reestablishes the values before you started editing, provided you have not yet pressed Apply.

Print Menu

Displays the choices for printing hardcopy output:

Current View, Day, Week, Month or Year. You can also get a hard copy list of the appointments and the Todo list for a Day, Week, Month or Year. For v3, multiple pages will be used to print appointments that do not fit on one page, and a * will be used to signify that further appointments follow on subsequent pages. with not enough You may set the printer settings in the Properties.

Prev Changes the display to the previous logical unit of calendar data.

Today Changes the display to today's logical unit of calendar data.

Next Changes the display to the following logical unit of calendar data.

FILES

/usr/spool/calendar/callog.\$USER
\$HOME/.desksetdefaults (for v3)
\$HOME/.cm.rc (for v2)
\$HOME/*.cal
/usr/openwin/bin/rpc.cmsd

SEE ALSO

rpc.cmsd(1), **cm_lookup(1)**, **cm_insert(1)**, and **cm_delete(1)** which are simple tty versions of **cm**.

Solaris User's Guide

"About Calendar Manager" in the Help Handbook available through the Help option on the Workspace menu.

| | |
|--------------------|---|
| NAME | cmap_alloc – allocate default colormaps for non-default visuals |
| SYNOPSIS | cmap_alloc [-display <i>display:n.screen</i>] [-force] [-allscreens] [-depth <i>n</i>] [-visual <i>class</i>] [-verbose] [-help] |
| DESCRIPTION | cmap_alloc creates empty colormaps for non-default dynamic visuals that are suitable for use as default colormaps. An X11 server has only one default colormap, and it is associated with the default visual. Clients that need to access non-default visuals have to create their own colormaps. This can lead to gratuitous colormap flashing if a lot of applications are running in non-default visuals. cmap_alloc will create colormaps that can be used by such applications as the default colormap for the given visual. Once the colormap has been created for a given visual, it will place the colormap's XID into the RGB_DEFAULT_MAP property on the root window of the display. Clients that need to access non-default dynamic visuals should search this property using XGetRGBColor-maps() for a suitable colormap to use. |
| OPTIONS | <p>-display <i>display:n.screen</i> Indicates which X11 server to use. The default is to use the value set in the DISPLAY environment variable.</p> <p>-force Create a colormap even if a suitable colormap is already listed in the RGB_DEFAULT_MAP property.</p> <p>-allscreens Create default colormaps for all screens of the display. Without this option, colormaps will only be created for the default screen.</p> <p>-depth <i>n</i> Create default colormaps only for those visuals that have depth <i>n</i>. The default is to create colormaps for visuals of all depths</p> <p>-visual <i>class</i> Create default colormaps only for those visuals that are of class <i>class</i>. Where <i>class</i> is one of the following: GrayScale, PsuedoColor, DirectColor</p> <p>-verbose Print out diagnostic messages useful for debugging.</p> <p>-help Print out a short usage message and exit.</p> |
| SEE ALSO | xstdcmap(1), cmap_compact(1) |
| DIAGNOSTICS | <p>can't open display ... Error in connecting to the X11 server. Check DISPLAY environment variable.</p> <p>unknown depth Invalid depth specified for the -depth command line option</p> <p>unknown visual class Invalid visual class specified for the -visual command line option</p> |

| | |
|--------------------|---|
| NAME | cmap_compact – colormap configuration utility to reduce colormap flashing. |
| SYNOPSIS | cmap_compact [<i>op</i>] [-display <i>dpyname</i>] |
| DESCRIPTION | <p>cmap_compact is a utility which allows certain colors to be designated as special and to be promoted to the high end of the default colormap (toward pixel 255). This reduces colormap flashing in many situations.</p> <p>This technique has no effect for monochrome screens; it applies only to color screens. For more information about the general problem of colormap flashing and instructions for using cmap_compact to solve the problem, refer to the OpenWindows documentation.</p> |
| OPTIONS | <p>-display <i>dpyname</i> Indicates the X11 display to use. ':0' is the default.</p> <p><i>op</i> can be one of the following operations:</p> <p>save Records the RGB color values of all shareable (read-only) colors in the default colormap for each active screen and saves these values on the file <code>~/.owcolors</code>. The black and white pixels (typically at pixels 254 and 255) are ignored during the save. Note that all colors are ignored for screens with a StaticColor default colormap. Note also that the server is grabbed during the save.</p> <p>init Reads the colors which have been saved in <code>~/.owcolors</code> and allocates them as read-only colors at the high-end of the default colormap. The color allocations for those applications which use these colors will be derived from these high colors. If there is no <code>~/.owcolors</code> file, nothing happens and no message is printed out.</p> <p>The colors are allocated in reverse order of the order in which they were saved. Thus, the lowest pixel in the colormap at save time becomes the highest pixel at initialization time. This operation is typically invoked from <code>~/.xinitrc</code> prior to starting the color applications. This operation creates on the root window of each screen a property named <code>XA_COMPACTED_COLORS_COUNT</code> of type <code>XA_INTEGER</code>. The value of this property specifies the number of initialized colors for that screen. Note: this value also includes the black and white pixels.</p> <p>An X11 <i>close down mode</i> of RetainPermanent is set so that the colors this program allocates will stay around after cmap_compact exits. These colors may be freed by calling cmap_compact dealloc . Before this program allocates its colors, it frees any colors left over from a previous invocation.</p> <p>If there are more saved colors than can fit in the default colormap, as many colors as will fit are allocated and a warning message is printed out. This situation, for example, might occur if the the default visual were switched from an 8-bit to a 4-bit visual without discarding saved colors.</p> |

Note: no allocations occur for screens with a StaticColor default colormap.

Note: The server is grabbed during the initialization.

discard

This operation removes the file `~/.owcolors`.

dealloc This operation frees colors allocated by a previous invocation of **cmap_compact init**.

show This operation prints out the RGB values of the saved colors for each screen.

**EFFECT ON THE
X11 SERVER**

The **Xsun(1)** server also reads the file `~/.owcolors` to determine the amount that it should shift down the color cube in the StaticColor colormap (this is called the *cube offset*).

Because there is only a single StaticColor colormap in the server, a decision must be made about how to handle different numbers of saved workspace colors on different screens.

The server currently takes the simplest approach: the cube offset is the number of saved colors for the screen with the largest number of colors.

The cube offset does not include the black and white pixels because these pixels are not shifted down with the rest of the cube; these pixels stay at the high end of the StaticColor colormap.

No matter what the value of the cube offset, the base pixel of the cube is always greater or equal to 0.

In addition to shifting the StaticColor color cube downward (toward 0) to avoid flashing with the saved colors, the black and white pixels which are allocated in the default colormap are moved to the high end of the default colormap (toward 255).

FILES

`~/.owcolors`

The file which is generated by **cmap_compact save**. Contains a list of saved, read-only workspace colors for all screens.

SEE ALSO

Xsun(1), **openwin(1)**

OpenWindows Server Programmer's Guide

BUGS

Currently does not support using the same `~/.owcolors` file for different machine architectures.

| | |
|--------------------|--|
| NAME | cmdtool – run a shell (or other program) in an OpenWindows enhanced terminal window |
| SYNOPSIS | cmdtool [-C] [-M bytes] [-P count] [-B boldstyle] [-I command] [<i>generic-tool-arguments</i>] [<i>program</i> [<i>program-arguments</i>]] |
| DESCRIPTION | <p>Cmdtool is the standard OpenWindows support facility for shells and other programs. When invoked, cmdtool runs a program (usually a shell) in a text-based command window. Characters typed on the keyboard are inserted at the caret. If the program is a shell, that shell accepts and runs commands in the usual manner. Cmdtool also supports programs that perform cursor motions directly, such as vi(1).</p> <p>The text of the current command line can be edited using normal textedit(1) functions. The command window displays a log of the session, which can be scrolled through using the scrollbar (unless the escape command mode is in effect). This log can be edited, and saved by choosing the ‘Store as New File’ item in the text facility’s pop-up menu.</p> <p>Cmdtool’s log file has a default wrap-around value of 100,000 bytes. When the size of the log file exceeds this value, text at the top of the file is lost as new text is added at the insertion point. The default value can be changed with the -M option or by placing the following line in the user’s ~/.Xdefaults file:</p> <p style="padding-left: 40px;">cmdtool.maxLogFileSize: <i>max_size</i></p> |
| OPTIONS | <p>-C Console cmdtool. Redirect system console output to this cmdtool. Display console messages in this cmdtool, which might otherwise appear in unexpected places on the workstation screen. Since a cmdtool window can be scrolled, console error messages can be recorded for later examination.</p> <p>-M bytes Set the log file /tmp/tty.txt.pid to wrap-around after the indicated number of <i>bytes</i> where the default value is 100,000 bytes. When the size of the log file exceeds this value, text at the beginning of the file is lost as new text is added at the insertion point. The log file contains editing characters as well as text so the number of characters in the cmdtool log itself may be less than the size of the log file. Also, the process size of the cmdtool is not limited to the indicated number of <i>bytes</i>.</p> <p>-P count Checkpoint the log after every set of <i>count</i> editing operations.</p> <p>-B boldstyle Set the style for displaying bold text to <i>boldstyle</i>. <i>boldstyle</i> can be a string specifying one of the choices for the term.boldstyle default, or it may be a numerical value for one of those choices, from 0 to 8, corresponding to the placement of the choice in the list.</p> <p>-I command Pass <i>command</i> to the shell. SPACE characters within the command must be escaped.</p> <p><i>generic-tool-arguments</i> cmdtool accepts the generic tool arguments listed in xview(7).</p> |

| | |
|--------------------------------|--|
| USAGE | <i>program</i> [<i>program-arguments</i>] |
| .Xdefaults File Options | If a <i>program</i> argument is present, cmdtool runs it and passes any remaining arguments to that <i>program</i> . If no <i>program</i> is given, cmdtool runs the program indicated by the SHELL environment variable, or /bin/sh by default. |
| cmdtool Windows | <p>You can specify a number of defaults using the options in the .Xdefaults file that effect the behavior of cmdtool. The ones of interest are those that begin with text, term, or key-board. See xview(7) for more detailed information.</p> <p>The window created by cmdtool is based on the text facility that is described in the textedit man page. The user is given a prompt at which to type commands and pop-up menus from which to select command options.</p> <p>cmdtool windows support cursor motions, using an /etc/terminfo entry called sun-cmd. Command windows automatically set the TERM environment variable to sun-cmd. So, if you rlogin(1) to a machine that does not have an entry for sun-cmd in its /etc/terminfo file, the error message "Type sun-cmd unknown" results. To rectify this, type the command set TERM=sun. Programs written using the curses(3X) library packages will work in a command window, but programs hard-coded for sun-type terminals may not. When supporting a program that performs cursor motions, the command window automatically takes on the characteristics of a tty window (as with shelltool(1)). When that program terminates or sleeps, the full command window functionality is restored.</p> <p>cmdtool supports programs that use CBREAK and NO ECHO terminal modes. This support is normally invisible to the user. However, programs that use RAW mode, such as rlogin(1) and script(1), inhibit command-line editing with the mouse. In this case, however, tty-style ERASE, word-kill and line-kill characters can still be used to edit the current command line.</p> |
| The cmdtool Menu | <p>The cmdtool window menu is called the Term Pane menu and contains the following options and their submenus:</p> <p>History Creates a a list of commands used during the cmdtool session.</p> <p>Mode</p> <ul style="list-style-type: none"> Editable You can edit the contents of the window. Read Only You can only read from the window. Store Log as new file Create a new file that contains the contents of the log. Clear log Clears all entries from the log. <p>Edit Provides a set of editing functions for this window.</p> <ul style="list-style-type: none"> Again Repeats the last action. Undo <ul style="list-style-type: none"> Undo Last Edit Undoes the last edit made in cmdtool. Undo All Edits |

- Undoes all edits made during this session of **cmdtool** .
- Copy** Makes a copy of the selected text and places it on the clipboard.
- Paste** Pastes a copy of the text stored on the clipboard at the cursor location.
- Cut** Deletes the selected text.
- Find** Provides a set of find and replace functions.
- Find and Replace**
Brings up a pop-up menu containing text fields and command buttons that allow you to search forward and backward through the file being edited for specific text strings. Allows you to specify options for the replacement of text.
- Find Selection**
Forward Searches forward to find a selected text string.
Backward Searches backward to find a selected text string.
- Find Marked Text**
Brings up a pop-up menu that allows you to find text that is included between specified bracket pairs. Also allows you to insert or remove bracket pairs around selected text.
- Replace** |>field<| >
Allows you to replace selected text forward and backward throughout the file.
- Extras** A user-definable pull-right menu controlled by the `/usr/openwin/lib/locale/C/xview/.text_extras_menu` file. This can be overridden in two ways:
1) Change the value of the `.Xdefaults` parameter `text.extrasMenuFilename` to the correct file path.
2) Set the environment variable `EXTRASMENU` to the file desired.
Note that option 1 overrides option 2 if both are used. For more information see the OpenWindows user documentation.
- File Editor**
Enable Allows you to edit files from within **cmdtool**.
Disable Turns off the ability to edit files from within **cmdtool**.
- Scrolling**
Enables Scrolling
Enables scrolling within **cmdtool**.
Disable Scrolling
Turns off the ability to scroll within **cmdtool**. Once scrolling in **cmdtool** is disabled, its functionality is identical to **shelltool** and a more restricted menu appears. Selecting the **Enable Scrolling** option from the restricted menu restores the full menu and functionality of **cmdtool**.

User Defined
Keyboard
Remapping

| | |
|---------------------|---|
| | The file <code>~/textswrc</code> specifies filter programs that are assigned to (available) function keys. These filters are applied to the contents of the primary selection. Their output is entered at the caret. |
| Accelerators | Text facility accelerators that are especially useful in command windows are described here. See <code>textedit(1)</code> for more information. CTRL-RETURN Position the caret at the bottom, and scroll it into view as determined by <code>Text.LowerContext</code> . CAPS-lock Toggle between all-upper-case keyboard input, and mixed-case. |
| FILES | <code>/tmp/tty.txt.pid</code> log file <code>~/textswrc</code> <code>~/ttyswrc</code> <code>/usr/openwin/lib/locale/C/xview/.text_extras_menu</code> <code>\$HOME/.TextExtraMenu</code> <code>/etc/terminfo</code> <code>/usr/bin/sh</code> |
| SEE ALSO | <code>rlogin(1)</code> , <code>script(1)</code> , <code>shelltool(1)</code> , <code>textedit(1)</code> , <code>vi(1)</code> , <code>xview(7)</code> , <code>curses(3X)</code> <i>Solaris User's Guide</i> |
| BUGS | Typing ahead while <code>cmdtool</code> changes between its scrollable and cursor motion modes will sometimes freeze <code>cmdtool</code> . Full terminal emulation is not complete. Some manifestations of this deficiency are: <ul style="list-style-type: none"> • File completion in the C shell does not work. • Enhanced display of text is not supported. |

| | |
|--------------------|---|
| NAME | cm_delete – delete appointments from Calendar Manager database |
| SYNOPSIS | cm_delete [-c <i>calendar</i>] [-d <i>date</i>] [-v <i>view</i>] |
| DESCRIPTION | The cm_delete utility is a tty interface to Calendar Manager cm (1). It can be used to delete appointments from the cm database via the RPC daemon rpc.cmsd (1). Appointments are deleted one at a time. Each of the components of an appointment is specified using one of the command line flags followed by the desired value. The current list of appointments for the specified date (see <i>date</i> , <i>view</i> options) is displayed, numbered sequentially starting with 1. User is prompted for the number to delete. Once an appointment is deleted, the list of remaining appointments is redisplayed. At this point the user may specify another number, or just <return> to quit. |
| OPTIONS | <p>-c <i>calendar</i> The name of the target calendar. Calendar names take the form "user@host", where the user is a user's login name and the host is the host machine name. An example is "felix@cat". If no target calendar is specified, calendar defaults to the current user at the current host machine.</p> <p>-d <i>date</i> The deletion date for the appointment. The date is specified using the form "mm/dd/yy", although certain other references such as "today", "Tuesday", "tomorrow", etc. are correctly calculated. If no date is specified, date defaults to today's date.</p> <p>-v <i>view</i> View span. This controls the span of time to display. The user may specify "day", "week", or "month". The "day" view displays all appointments for the given date (see -d option above). The "week" view displays the full week which contains the given date, starting with Sunday. The "month" view displays the entire month which contains the given date, starting with the first of the month. The default view is "day".</p> |
| USAGE | The various components of an appointment are specified using command line flags followed by the desired value. Values may have embedded spaces, punctuation, etc., although quotes may be needed around strings which contain certain characters to protect them from interpretation by the local shell (e.g. /bin/csh). An argument is taken to begin at the first space after a flag and to continue until the first "-" after a space. |
| EXAMPLE | <p>The simplest form of cm_delete has no arguments:</p> <pre> example% cm_delete Appointments for Tuesday September 25, 1990: 1) Appointment 2) 10:30am-10:45am Morning Tea 3) 2:00pm-3:00pm Staff meeting 4) 4:30pm-5:30pm Phone home </pre> |

Item to delete (number)? 2

Appointments for Tuesday September 25, 1990:

- 1) Appointment
- 2) 2:00pm-3:00pm Staff meeting
- 3) 4:30pm-5:30pm Phone home

Item to delete (number)?

example%

To delete at a specific date:

example% cm_delete -d 09/26/90

Appointments for Wednesday September 26, 1990:

- 1) 11:00am-12:00pm Appointment
- 2) 11:30am-12:30pm Group Lunch
- 3) 4:00pm-5:00pm Tech Interview

Item to delete (number)? 1

Appointments for Wednesday September 26, 1990:

- 1) 11:30am-12:30pm Group Lunch
- 2) 4:00pm-5:00pm Tech Interview

Item to delete (number)?

example%

To delete from a specific target calendar:

example% cm_delete -c felix@cat

Appointments for Tuesday September 25, 1990:

- 1) Appointment
- 2) 10:15am-10:30am Coffee
- 3) 11:15am-11:30am Doughnuts
- 4) 2:00pm-2:15pm Coffee
- 5) 3:30pm-3:45pm Snack
- 6) 4:30pm-4:45pm Coffee

Item to delete (number)? 5

Appointments for Tuesday September 25, 1990:

- 1) Appointment
- 2) 10:15am-10:30am Coffee
- 3) 11:15am-11:30am Doughnuts
- 4) 2:00pm-2:15pm Coffee
- 5) 4:30pm-4:45pm Coffee

Item to delete (number)?

example%

To delete multiple appointments:

example% cm_delete

Appointments for Tuesday September 25, 1990:

- 1) Appointment
- 2) 10:15am-10:30am Coffee
- 3) 11:15am-11:30am Doughnuts
- 4) 2:00pm-2:15pm Coffee
- 5) 3:30pm-3:45pm Snack
- 6) 4:30pm-4:45pm Coffee

Item to delete (number)? 5

Appointments for Tuesday September 25, 1990:

- 1) Appointment
- 2) 10:15am-10:30am Coffee
- 3) 11:15am-11:30am Doughnuts
- 4) 2:00pm-2:15pm Coffee
- 5) 4:30pm-4:45pm Coffee

Item to delete (number)? 3

Appointments for Tuesday September 25, 1990:

- 1) Appointment
- 2) 10:15am-10:30am Coffee
- 3) 2:00pm-2:15pm Coffee
- 4) 4:30pm-4:45pm Coffee

Item to delete (number)?

example%

FILES /usr/spool/calendar/callog.username

/usr/etc/rpc.cmsd or \$OPENWINHOME/rpc.cmsd

SEE ALSO rpc.cmsd(1), cm_insert(1), cm_lookup(1), cm(1)

| | |
|--------------------|--|
| NAME | cm_insert – insert appointments into Calendar Manager database |
| SYNOPSIS | cm_insert [-c <i>calendar</i>] [-d <i>date</i>] [-s <i>start</i>] [-e <i>end</i>] [-v <i>view</i>] [-w <i>what</i>] |
| DESCRIPTION | The cm_insert utility is a tty interface to Calendar Manager cm (1). It can be used to add new appointments to the cm database via the RPC daemon rpc.cmsd . Appointments are added one at a time. Each of the components of an appointment is specified using one of the command line flags followed by the desired value. Once an appointment is added, the list of appointments for the specified date (see date , view options) is displayed. |
| OPTIONS | <p>-c <i>calendar</i> The name of the target calendar. Calendar names take the form "user@host", where the user is a user's login name and the host is the host machine name. An example is "felix@cat". If no target calendar is specified, calendar defaults to the current user at the current host machine.</p> <p>-d <i>date</i> The insertion date for the appointment. The date is specified using the form "mm/dd/yy", although certain other references such as "today", "Tuesday", "tomorrow", etc. are correctly calculated. If no date is specified, date defaults to today's date.</p> <p>-s <i>start</i> The starting time for the appointment. The time is specified using the form "hh:mm" plus an optional "am" or "pm" meridian. If "am/pm" is left off, "am" is assumed. Time specified using 24-hour convention (e.g. "15:30" instead of "3:30 pm") are acceptable and are converted to meridian time before insertion. If no starting time is specified, starting time defaults to "magic" time (see Calendar Manager manual) and no time appears next to the appointment.</p> <p>-e <i>end</i> The ending time for the appointment. The time is specified as above "starting time". It is considered an error to specify an ending time without specifying a starting time.</p> <p>-v <i>view</i> View span. This controls the span of time to display. The user may specify "day", "week", or "month". The "day" view displays all appointments for the given date (see -d option above). The "week" view displays the full week which contains the given date, starting with Sunday. The "month" view displays the entire month which contains the given date, starting with the first of the month. The default view is "day".</p> <p>-w <i>what</i> What the appointment is about. The user may specify a string of text describing the nature of the appointment. Up to 5 lines of text can be specified by placing "\n" (actual characters "\" and "n", not newline) between lines. It may be necessary to escape the "\" character ("\\n") or enclose string in quotes ("\"this string in quotes\"") to avoid interpretation by the local shell. If not specified, "what" defaults to "Appointment".</p> |

USAGE

The various components of an appointment are specified using command line flags followed by the desired value. Values may have embedded spaces, punctuation, etc., although quotes may be needed around strings which contain certain characters to protect them from interpretation by the local shell (e.g. /bin/csh). An argument is taken to begin at the first space after a flag and to continue until the first "-" after a space.

EXAMPLE

The simplest form of **cm_insert** has no arguments:

```
example% cm_insert
Appointments for Tuesday September 25, 1990:
  1) Appointment
```

```
example%
```

To insert at a specific time:

```
example% cm_insert -s 11:00 am
Appointments for Tuesday September 25, 1990:
  1) Appointment
```

```
  2) 11:00am-12:00pm Appointment
```

```
example%
```

To insert at a specific start and end time:

```
example% cm_insert -s 11:00 am -e 11:28 am
Appointments for Tuesday September 25, 1990:
  1) Appointment
```

```
  2) 11:00am-11:28am Appointment
```

```
example%
```

To insert at a specific time and date:

```
example% cm_insert -s 11:00 am -d 09/26/90
Appointments for Wednesday September 26, 1990:
  1) 11:00am-12:00pm Appointment
```

example%

To insert at a specific time, date, and message:

```
example% cm_insert -s 11:00 am -d 09/26/90 -w call home  
Appointments for Wednesday September 26, 1990:  
1) 11:00am-12:00pm Appointment  
2) 11:00am-12:00pm call home
```

example%

Lastly, for multiple line appointments:

```
example% cm_insert -s 12:00 -w call dentist\\n  
no thanks\\ncancel appointment  
Appointments for Tuesday September 25, 1990:  
1) Appointment  
  
2) 11:00am-12:00pm Appointment  
  
3) 12:00pm-1:00pm call dentist  
no thanks  
cancel appointment  
example%
```

FILES /usr/spool/calendar/callog.username
/usr/etc/rpc.cmsd or \$OPENWINHOME/rpc.cmsd

SEE ALSO rpc.cmsd(1), cm_delete(1), cm_lookup(1), cm(1)

| | |
|--------------------|---|
| NAME | cm_lookup – look up appointments from Calendar Manager database |
| SYNOPSIS | cm_lookup [<i>-c calendar</i>] [<i>-d date</i>] [<i>-v view</i>] |
| DESCRIPTION | The cm_lookup utility is a tty interface to Calendar Manager cm(1) . It can be used to look up appointments from the cm database via the RPC daemon rpc.cmsd(1) . Each of the components of an appointment is specified using one of the command line flags followed by the desired value. The current list of appointments for the specified date (see date, view options) is displayed. |
| OPTIONS | <p>-c calendar The name of the target calendar. Calendar names take the form "user@host", where the user is a user's login name and the host is the host machine name. An example is "felix@cat". If no target calendar is specified, calendar defaults to the current user at the current host machine.</p> <p>-d date The date for the appointment. The date is specified using the form "mm/dd/yy", although certain other references such as "today", "Tuesday", "tomorrow", etc. are correctly calculated. If no date is specified, date defaults to today's date.</p> <p>-v view View span. This controls the span of time to display. The user may specify "day", "week", or "month". The "day" view displays all appointments for the given date (see -d option above). The "week" view displays the full week which contains the given date, starting with Sunday. The "month" view displays the entire month which contains the given date, starting with the first of the month. The default view is "day".</p> |
| USAGE | The various components of an appointment are specified using command line flags followed by the desired value. Values may have embedded spaces, punctuation, etc., although quotes may be needed around strings which contain certain characters to protect them from interpretation by the local shell (e.g. /bin/csh). An argument is taken to begin at the first space after a flag and to continue until the first "-" after a space. |
| EXAMPLE | <p>The simplest form of cm_lookup has no arguments:</p> <pre> example% cm_lookup Appointments for Tuesday September 25, 1990: 1) Appointment 2) 10:30am-10:45am Morning Tea 3) 2:00pm-3:00pm Staff meeting 4) 4:30pm-5:00pm Phone home example%</pre> |

To look up a specific date:

```
example% cm_lookup -d 09/26/90
Appointments for Wednesday September 26, 1990:
  1) 11:00am-12:00pm Appointment
  2) 11:30am-12:30pm Group Lunch
  3) 4:00pm-5:00pm Tech Interview
```

```
example%
```

To look up a specific target calendar:

```
example% cm_lookup -c felix@cat
Appointments for Tuesday September 25, 1990:
  1) Appointment
  2) 10:15am-10:30am Coffee
  3) 11:15am-11:30am Doughnuts
  4) 2:00pm-2:15pm Coffee
  5) 3:30pm-3:45pm Snack
  6) 4:30pm-4:45pm Coffee
```

```
example%
```

To look up an entire week's appointments:

```
example% cm_lookup -v week
Appointments for Sunday September 23, 1990:
  1) 6:00am-5:00pm Hiking
```

```
Appointments for Monday September 24, 1990:
  1) 11:00am-11:30am Sync with East Coast
  2) 4:00pm-4:15pm Confirm flight
```

```
Appointments for Tuesday September 25, 1990:
  1) Appointment
  2) 10:15am-10:30am Coffee
  3) 11:15am-11:30am Doughnuts
  4) 2:00pm-2:15pm Coffee
  5) 3:30pm-3:45pm Snack
  6) 4:30pm-4:45pm Coffee
```

```
Appointments for Wednesday September 26, 1990:
```

- 1) 11:00am-11:15am Appointment
- 2) 11:30am-12:30pm Group Lunch
- 3) 4:00pm-5:00pm Tech Interview

Appointments for Friday September 28, 1990:

- 1) Documentation
- 2) 10:00am-11:00am Staff meeting

Appointments for Saturday September 29, 1990:

- 1) 9:00am-11:00am Raquetball with Debbie

example%

Notice that "Thursday" does not appear, since there were no appointments on that day.

FILES /usr/spool/calendar/callog.username
/usr/etc/rpc.cmsd or \$OPENWINHOME/rpc.cmsd

SEE ALSO rpc.cmsd(1), cm_insert(1), cm_delete(1), cm(1)

| | |
|--------------------|---|
| NAME | CoEd – a ToolTalk-based shared editor |
| SYNOPSIS | CoEd [file] |
| DESCRIPTION | <p>CoEd a shared text editor built with DevGuide, the OLIT textedit widget, and libCoEd. libCoEd is a C++ library implementing a ToolTalk protocol that can turn any editor into a shared editor, using the dOPT algorithm. dOPT allows multiple users to edit the same text at the same time, without locking, and with all locally-generated edits being applied immediately to the local text. Being merely a demo, CoEd has the following limitations. CoEd has no facility for saving the edited file to disk.</p> <p>Each instance of CoEd must be started before any editing occurs. We have not implemented the part of the protocol that allows newcomers to join an ongoing editing session. However, any CoEd can quit at any time without disrupting the others. See the README file with the CoEd source code for a sketch of the rest of the protocol.</p> |
| FILES | <p>/.global_coed_file If no file is named on the command line, all instances of CoEd on the same machine will rendezvous on this file.</p> |
| SEE ALSO | <p>ttsession(1) C.A. Ellis and S.J. Gibbs, "Concurrency Control in Groupware Systems," Proc. 1989 ACM SIGMOD, Int. Conf. on the Mgt. of Data</p> |
| WARNINGS | <p>Building CoEd requires DevGuide, C++, and ToolTalk (which is part of OpenWindows). CoEd is known to build with DevGuide 3.0.1, C++ from SPARCompilers 2.0.1, and either OpenWindows 3.1 or OpenWindows 3.2.</p> <p>The libgolit in DevGuide 3.0.1 was based on the OW3.0.1/3.1 OLIT libraries, and has some incompatibilities with the OW3.2 OLIT libraries. A CoEd built with libgolit 3.0.1 and running under OW3.2 will crash, e.g., if you press MENU or HELP over the OLIT textedit widget embedded in CoEd. However, the basic features of editing, selecting, copy/paste, drag-and-drop, and scrolling seem to work.</p> |

| | |
|---------------------|--|
| NAME | colorchooser – change icon colors in deskset tools |
| SYNOPSIS | colorchooser |
| AVAILABILITY | The Color Chooser is available with the OpenWindows environment on color workstations. |
| DESCRIPTION | colorchooser is an XView-based OpenWindows tool that lets users change the colors of icons used by DeskSet applications. Currently, the Color Chooser can be invoked from the Binder and Icon Editor applications. |
| USAGE | <p>The Color Chooser can be used with the Binder to specify foreground and background colors for the icons displayed by the File Manager or other DeskSet applications. To display the Color Chooser from the Binder, click SELECT on the Foregr Color or Backgr Color menu button. (These are the buttons to the right of the text fields). When the Color Chooser is displayed, the icon image with the currently Foreground and Background colors is displayed in the upper left corner of the palette. To use any of the colors displayed in the palette for the Binder, click SELECT on the desired color in the Color Chooser palette then click SELECT on the Apply button to record the color change.</p> <p>The Color Chooser can also be used with the Icon Editor to create a multi-color icon. To display the Color Chooser from the Icon Editor, click SELECT on the button labeled "Palette". The Color Chooser can then be used to specify the color to draw with. The current color that Icon Editor will use for drawing is displayed in the upper left corner of the color palette. To use any of the colors displayed in the palette for the Icon Editor, click SELECT on the desired color in the Color Chooser palette then click SELECT on the Apply button to record the color change. The pointer changed to be the selected color as a reminder of the color you are presently using.</p> |
| FILES | /usr/openwin/share/xnews/client/ds_server_init/ds_colors.txt This file contains the colors displayed on the color palette of the colorchooser . |
| SEE ALSO | ds_server_init(1), iconedit(1), binder(1) <i>Solaris User's Guide</i> |

| | | | | | | | | | | | | | | | |
|--------------------|--|-----|---|-----|---|-----|---|----|-----------------------------------|----|------------------------------------|----|-----------------------------|-----|----------------------------------|
| NAME | constype – print type of Sun console | | | | | | | | | | | | | | |
| SYNOPSIS | constype | | | | | | | | | | | | | | |
| DESCRIPTION | constype prints on the standard output the Sun code for the type of display that the console is. It is one of: <table><tr><td>bw?</td><td>Black and White, where ? is 1-4. (eg) 3-50s are bw2</td></tr><tr><td>cg?</td><td>Colour Graphics display, where ? is 1-4</td></tr><tr><td>gp?</td><td>Optional Graphics Processor board, where ? is 1-2</td></tr><tr><td>gx</td><td>Sun GX (cg6) Graphics Accelerator</td></tr><tr><td>gs</td><td>Sun GS (cg12) Graphics Accelerator</td></tr><tr><td>gt</td><td>Sun GT Graphics Accelerator</td></tr><tr><td>ns?</td><td>Not Sun display - where ? is A-J</td></tr></table> <p>This is useful in determining startup values and defaults for window systems.</p> | bw? | Black and White, where ? is 1-4. (eg) 3-50s are bw2 | cg? | Colour Graphics display, where ? is 1-4 | gp? | Optional Graphics Processor board, where ? is 1-2 | gx | Sun GX (cg6) Graphics Accelerator | gs | Sun GS (cg12) Graphics Accelerator | gt | Sun GT Graphics Accelerator | ns? | Not Sun display - where ? is A-J |
| bw? | Black and White, where ? is 1-4. (eg) 3-50s are bw2 | | | | | | | | | | | | | | |
| cg? | Colour Graphics display, where ? is 1-4 | | | | | | | | | | | | | | |
| gp? | Optional Graphics Processor board, where ? is 1-2 | | | | | | | | | | | | | | |
| gx | Sun GX (cg6) Graphics Accelerator | | | | | | | | | | | | | | |
| gs | Sun GS (cg12) Graphics Accelerator | | | | | | | | | | | | | | |
| gt | Sun GT Graphics Accelerator | | | | | | | | | | | | | | |
| ns? | Not Sun display - where ? is A-J | | | | | | | | | | | | | | |
| BUGS | Not tested on all monitor types | | | | | | | | | | | | | | |
| COPYRIGHT | Copyright 1988, SRI | | | | | | | | | | | | | | |
| AUTHOR | Doug Moran <moran@ai.sri.com> | | | | | | | | | | | | | | |

| | |
|---------------------|---|
| NAME | convert_to_Xdefaults – convert SunView1 defaults into equivalent Xdefaults |
| SYNOPSIS | convert_to_Xdefaults [<i>filename</i>] |
| AVAILABILITY | Available with the <i>OpenWindows Application Environment</i> . For information about installing OpenWindows, refer to the current Solaris system configuration and installation guide. |
| DESCRIPTION | convert_to_Xdefaults is a shell script which uses sed(1) scripts to convert <i>SunView1.x</i> defaults to <i>X Window</i> (Xdefaults) defaults. convert_to_Xdefaults reads <i>filename</i> , converting SunView1 defaults into their equivalent Xdefaults for XView. Defaults that are no longer supported or are not recognized as standard <i>SunView1</i> defaults are commented out with a '!' (exclamation point) at the beginning of the default entry. The output of conversion is directed to standard output (stdout). The defaults file used should be located in your <i>\$HOME</i> directory and should be named <i>.Xdefaults</i> . |
| EXAMPLES | <p>For an example of how this command works, run the script on your <i>Sunview1</i> defaults file:</p> <pre>convert_to_Xdefaults \$HOME/.defaults > \$HOME/.Xdefaults</pre> <p>When the script is complete, edit the resulting file (<i>\$HOME/.Xdefaults</i>) and remove any unconverted entries and/or make modifications suggested in the file by the script. Remove the comments and the ! sign.</p> |
| SAMPLES | <p>SunView1 defaults entries ...</p> <pre> /Text/Auto_indent "True" /Text/Extras_menu_filename "/home/blinky/bob/.text_extras_menu" /Scrollbar/Thickness "20" /Mail/Set/folder "/home/blinky/bob/mail_folder" /Text/Multi_click_timeout "100" </pre> <p>are converted to the Xdefault...</p> <pre> Text.AutoIndent: True Text.ExtrasMenuFilename: /home/blinky/bob/.text_extras_menu !/Scrollbar/Thickness 20 !/Mail/Set/folder /home/blinky/bob/mail_folder !OpenWindows.MultiClickTimeout: ! (now in tenths of seconds rather than millisecs) 100 </pre> |

Note that the **/Scrollbar/Thickness** and **/Mail/Set/folder** entries were NOT converted, but were left in the file as comments. Whenever possible, instructions are included in the file for discrepancies between the two types of defaults. For instance, the comment, "now in tenths of seconds" is useful information about the value of the **OpenWindows.MultiClickTimeout** default. Comments and instructions should both be completely removed from the file.

FILES **\$OPENWINHOME/bin/convert_to_Xdefaults** where **\$OPENWINHOME** is the installation/mount point for XView (**/usr/openwin** by default).

SEE ALSO **sed(1), textedit(1), vi(1)**

| | |
|---------------------|--|
| NAME | convert_to_xview – convert a SunView1 source program to XView source |
| SYNOPSIS | convert_to_xview [-m] <i>filename</i> .. |
| AVAILABILITY | This command is available with the XView software distribution. |
| DESCRIPTION | <p>convert_to_xview is a shell script which uses sed(1) scripts to convert SunView programs to the XView Application Programming Interface (API). Convert_to_xview parses <i>filename</i> and creates a new file with the XView API in the current directory called <i>filename.converted_to_xview</i>. The default conversion that is done is called FULL conversion. FULL conversion of SunView source converts everything to XView naming conventions regardless of API compatibility support (e.g., changes WIN_FONT to XV_FONT even though WIN_FONT would still work).</p> <p>The other type of conversion is called MINIMAL conversion. MINIMAL conversion retains SunView compatibility wherever possible and inserts a unique flag and comments at every instance where manual conversion is necessary in C language source comment form. The flag and comments will look something like this:</p> <pre style="margin-left: 40px;">#ifdef XVIEW_COMMENT XView CONVERSION - Make sure to use xv_init to process the attrs first. Sect 3.2 #endif</pre> <p>The original SunView file is not modified. After the file is converted, you should then search for</p> <pre style="margin-left: 40px;">XView CONVERSION</pre> <p>in the new converted program (<i>filename.converted_to_xview</i>). Use the conversion documentation, <i>XView 3.1: Converting SunView Applications</i>, to determine the proper conversion for these flagged items. In some possible cases, the comments may make references to sections in the manual which should be consulted to correctly convert something.</p> |
| OPTIONS | -m Perform minimal conversion only. |
| ENVIRONMENT | The script recognizes the environment variable \$OPENWINHOME as the root directory for the installation point for convert_to_xview . By default it should be installed into the root directory '/' . Additionally, the sed (1) scripts that are used by convert_to_xview must be located in the \$OPENWINHOME/bin directory. |
| EXAMPLES | <p>Convert foo.c from SunView1 to XView:</p> <pre style="margin-left: 40px;">% convert_to_xview foo.c ---Converting File: foo.c --Done %</pre> |

Now go in and edit (with your favorite text editor such as **vi**, **textedit**, etc.) the result of the conversion (`my_program.c.converted_to_xview`) and see if there is anything that didn't get converted:

```
% textedit foo.c.converted
```

Do only minimal conversion of `my_program.c` & `your_program.c` to XView:

```
% convert_to_xview -m foo.c blah.c  
---Converting File: foo.c  
---Converting File: blah.c  
--Done  
%
```

The above would create two files new files and each will only had minimal conversion performed (just flags inserted).

FILES

`$/OPENWINHOME/bin/convert_to_xview`

Where `$/OPENWINHOME` is the installation/mount point for XView.

SEE ALSO

`sed(1)`, `textedit(1)`, `vi(1)`, `sh(1)`

| | |
|---------------------|---|
| NAME | docviewer – OpenWindows viewer for AnswerBook on-line documentation |
| SYNOPSIS | docviewer -d <i>document-name</i> -p <i>tooltalk-procid</i> [-c <i>card-catalog-file</i>] |
| AVAILABILITY | SUNWowrqd |
| DESCRIPTION | <p>docviewer and navigator(1) constitute an OpenWindows application for viewing and navigating AnswerBook on-line document collections.</p> <p>docviewer displays PostScript files found using navigator. docviewer lets you page through a document, print pages, magnify or reduce pages, and follow hypertext links and cross-references to other documents.</p> <p>You should not start docviewer directly. It is started automatically by navigator as needed. If you are running several navigators simultaneously, they will each start their own docviewers.</p> |
| OPTIONS | <p>docviewer accepts most of the generic tool arguments described in xview(7), as well as the following options:</p> <p>-d <i>document-name</i> Specify the name of the document to be viewed.</p> <p>-c <i>card-catalog</i> Specify the name of the card catalog file used to locate AnswerBooks. See ab_cardcatalog(4) for more information.</p> <p>-p <i>tooltalk-procid</i> Specify the tooltalk process id of the navigator invoking this docviewer.</p> |
| USAGE | <p>Once a document is displayed in the docviewer window, you can view it as follows:</p> <p>Page Turning Click SELECT on the left and right arrow buttons at the top of the docviewer window to page back and forth through a document. The PgUp and PgDn keys on the keyboard perform the same functions. In addition, the Home and End keys let you page through a document one chapter at a time.</p> <p>Page History docviewer keeps track of each page visited. Click SELECT on the Go Back button until the window returns to the page you want to revisit, or use the Undo keyboard function.</p> <p>Hypertext Links docviewer has a simple hypertext mechanism. Hypertext links are displayed as rectangular outlines around words or graphics on a page. Double-clicking SELECT on a link causes docviewer to display the document and page to which the link points. Documents may have links to other documents within the same collection. Certain types of hypertext links can also initiate system processes. Double-clicking on these links will start up a program or shell script.</p> |

Page Magnification

Magnify or reduce the size of the document by pulling the resize corners on the **docviewer** window, or select the "View->Custom Magnification" menu item. Select the "View->Standard Magnification" menu item to reset the document to the standard size. For very high magnifications, select the "View->Partial Page" menu item so that the **docviewer** window still fits on the screen.

Printing

Click SELECT on the Print button to show the pop-up Print Window. In the Print Window, select the page or document to print, number of copies, and printing device (or file).

Page Information

Select the "Viewer->Page Info" menu item to show a pop-up window describing the chapter, book, and AnswerBook you are currently viewing.

ENVIRONMENT**AB_CARDCATALOG**

Specify the name of the card catalog file used to locate AnswerBooks. See **ab_cardcatalog(4)** for more information.

SEE ALSO

navigator(1), **viewprint(1)**, **answerbook(1)**, **ab_admin(1)**, **ab_cardcatalog(4)**,

DIAGNOSTICS

docviewer(1) displays PostScript error messages in the console.

BUGS

The AnswerBook document naming syntax used by **docviewer** is not documented.

NOTES

PostScript is a trademark of Adobe Systems Incorporated.
AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc.

| | |
|---|--|
| NAME | dps – Display PostScript imaging for the X Window System |
| DESCRIPTION | <p>This manual page provides information about the Display PostScript system, implemented as an extension to the X Window System.</p> <p>The INTRODUCTION section contains a brief, nontechnical description of the Display PostScript system.</p> <p>The remaining sections provide the application developer with more detailed technical information about the architecture.</p> <p>The REFERENCES section describes additional documentation and tells you how to use Adobe's public access file server.</p> |
| INTRODUCTION | <p>The PostScript language is a simple interpretive programming language with powerful graphics capabilities. Its primary purpose is to describe the appearance of text, graphical shapes, and images on printed or displayed pages. If an application, such as a word processing system or graphics package, produces a page description using the PostScript language, you can print the pages on a wide variety of PostScript printers and view them on monitors where the Display PostScript system is available.</p> <p>The Display PostScript system is a high-performance implementation of the PostScript language for interactive computer displays. The use of the Display PostScript system ensures true WYSIWYG (What You See Is What You Get) between the display and any PostScript printer.</p> |
| DISPLAY POSTSCRIPT SYSTEM ARCHITECTURE | <p>The Display PostScript system is part of the X Window System and is implemented as an X extension. Display PostScript applications use window system features for window placement and sizing, menu creation, and event handling, while using Display PostScript features to take care of imaging inside the window.</p> <p>Display PostScript system components include:</p> <ul style="list-style-type: none"> The PostScript interpreter. The Client Library – a C language interface to the basic facilities of the Display PostScript system. <i>pswrap</i> – a preprocessor that prepares PostScript language programs for invocation from a C program. <p>These components are discussed below.</p> |
| APPLICATION BUILDING BLOCKS | <p>Most of a Display PostScript application is written in C or another high-level language. It calls Client Library procedures to start a PostScript execution context, send programs and data to the PostScript interpreter, and get results from the interpreter. The Client Library is the application's primary interface to the Display PostScript system.</p> <p>In addition, it calls <i>wraps</i> – custom PostScript language procedures developed specifically for the application. Wraps are generated by the <i>pswrap</i> translator from application-specific PostScript language code.</p> |

USING PSWRAP

pswrap is a preprocessor that takes PostScript language code as input and embeds it in C-callable procedures, or wraps. The output of *pswrap* is compiled and linked with the rest of your application, which can then call the wraps to transmit PostScript language code to the PostScript interpreter.

A Display PostScript application uses C or another high-level language to perform calculations, communicate with the window system, read and write files, and do other application processing. It uses wraps primarily for imaging tasks.

Consider a procedure, **PSWDisplayText**, that places text on the screen at a particular *x,y* coordinate. A call to this wrap from the application program might look something like this:

```
PSWDisplayText(72.0, 100.0, "Hello World");
```

The body of the **PSWDisplayText** procedure is actually written in the PostScript language. It was defined to *pswrap* as follows:

```
defineps PSWDisplayText(float X,Y; char *text)
    X Y moveto
    (text) show
endps
```

In the wrap definition, the **defineps** and **endps** keywords tell *pswrap* where a given PostScript language program begins and ends. The **defineps** statement defines the resulting procedure call. The *pswrap* translator processes this input and produces a C language source-code file. When compiled and linked with the application, the **PSWDisplayText** procedure sends a PostScript language program to the interpreter (binary-encoded for more efficient processing), causing "Hello World" to be displayed on the screen.

See the *Programming the Display PostScript System with X* for further information.

THE CLIENT LIBRARY

The Display PostScript Client Library is a linkable library of compiled C procedures that provides an interface between the application and the Display PostScript system. It creates an environment for handling imaging calls to specific Client Library procedures like **DPSmoveto** and to custom wraps written for the application.

To the application programmer, it appears that Client Library procedures directly produce graphical output on the display. In fact, these procedures generate PostScript language statements and transmit them to the PostScript interpreter for execution; the PostScript interpreter then produces graphical output that is displayed by device-specific procedures in the Display PostScript system. In this way, the Client Library makes the full power of the PostScript interpreter and imaging model available to a C language program.

The Client Library includes procedures for creating, communicating with, and destroying PostScript execution contexts. A context consists of all the information (or "state") needed by the PostScript interpreter to execute a PostScript language program. In the Client

Library interface, each context is represented by a **DPSContextRec** data structure. PostScript execution contexts are described in the *PostScript Language Reference Manual, Second Edition*.

REFERENCES

Information about the PostScript Language and the Display PostScript system is available in a number of manuals and via the public access file server described below.

POSTSCRIPT LANGUAGE MANUALS

If you're new to the PostScript language, you should first read the following manuals (published by Addison-Wesley and available from Adobe Systems Incorporated or through your technical bookstore):

PostScript Language Reference Manual, Second Edition

The standard reference for the PostScript language. Describes the PostScript imaging model and the concepts and facilities of the PostScript interpreter. Documents the PostScript language. Required reading.

PostScript Language Tutorial and Cookbook

Introduction to the PostScript language in an informal, interactive style. Contains a collection of example programs that illustrate the PostScript imaging model.

PostScript Language Program Design

Guidelines for the advanced developer to use in designing and debugging PostScript language programs. Printer-oriented, but most of the information is relevant to writing a Display PostScript application.

DISPLAY POSTSCRIPT MANUALS

Once you're up to speed in the PostScript language, read *Programming the Display PostScript System with X*, available from Addison-Wesley. This book is collection of manuals that explain how to render text and graphics with the Display PostScript extension to X. It contains the following manuals:

Programming Guide

Explains how to render text and graphics with the Display PostScript extension to X.

Client Library Reference Manual

Describes the procedural interface to the Display PostScript system. Tells how to send programs and data to a PostScript execution context, how to handle context output, how to create and terminate a context. Contains procedure definitions, programming tips, and a sample application program.

Client Library Supplement for X

Describes Display PostScript features that are specific to the X Window System, such as context creation and additional error codes.

pswrap Reference Manual

Describes how to define C-callable procedures that contain PostScript language programs. Tells how to declare input arguments and output to be received from the interpreter. Documents the pswrap command line options.

Display PostScript Toolkit for X

Describes the Display PostScript Toolkit for the X Window System. It also contains information about locating PostScript language resources and about the *makepsres* utility.

**THE PUBLIC
ACCESS FILE
SERVER**

Adobe Systems Incorporated provides a public access file server. If you have access to Internet or UUCP electronic mail, you can use the public access file server to obtain the following information:

- Display PostScript system manuals
- Code examples
- AFM files
- Documentation updates

The public access file server is a mail-response program. That is, you send it a request by electronic mail and it mails back a response. (The "Subject:" line is treated as part of the message by the file server.)

To send mail to the file server, use one of the following addresses:

Internet **ps-file-server@adobe.com**

UUCP **...!decwd!adobe!ps-file-server**

To receive a quick summary of file server commands, send the following message:

help

To receive detailed information on how to use the file server, send the following message:

send Documents long.help

COLORMAP USAGE

The Display PostScript system uses entries from the default X colormap to display colors and grey values. You can configure this usage. Giving the Display PostScript system more colormap entries improves the quality of its rendering, but leaves fewer entries available to other applications since the default colormap is shared.

Resources in your *.Xdefaults* file control the colormap usage. Each resource entry should be of the form

DPSColorCube.visualType.depth.color: size

where

visualType is one of *GrayScale*, *PseudoColor*, or *DirectColor*.

depth is 1, 2, 4, 8, 12, or 24 and should be the largest depth equal to or less than the default depth.

color is one of the strings "reds", "greens", "blues", or "grays".

size is the number of values to allocate of that color.

These resources are not used for the static visual types *StaticGray*, *StaticColor*, or *TrueColor*. Specifying 0 for reds directs the Client Library to use only a gray ramp. This specification is particularly useful for gray-scale systems that incorrectly have

PseudoColor as the default visual.

For example, to configure a 5x5x4 color cube and a 17-element gray ramp for an 8-bit PseudoColor screen, specify these resources:

```
DPSColorCube.PseudoColor.8.reds: 5
DPSColorCube.PseudoColor.8.greens: 5
DPSColorCube.PseudoColor.8.blues: 4
DPSColorCube.PseudoColor.8.grays: 17
```

These resources use 117 colormap entries, 100 for the color cube and 17 for the gray ramp. For the best rendering results, specify an odd number for the gray ramp.

Resources that are not specified take these default values:

```
DPSColorCube.GrayScale.4.grays: 9
DPSColorCube.GrayScale.8.grays: 17

DPSColorCube.PseudoColor.4.reds: 2
DPSColorCube.PseudoColor.4.greens: 2
DPSColorCube.PseudoColor.4.blues: 2
DPSColorCube.PseudoColor.4.grays: 2
DPSColorCube.PseudoColor.8.reds: 4
DPSColorCube.PseudoColor.8.greens: 4
DPSColorCube.PseudoColor.8.blues: 4
DPSColorCube.PseudoColor.8.grays: 9
DPSColorCube.PseudoColor.12.reds: 6
DPSColorCube.PseudoColor.12.greens: 6
DPSColorCube.PseudoColor.12.blues: 5
DPSColorCube.PseudoColor.12.grays: 17

DPSColorCube.DirectColor.12.reds: 6
DPSColorCube.DirectColor.12.greens: 6
DPSColorCube.DirectColor.12.blues: 6
DPSColorCube.DirectColor.12.grays: 6
DPSColorCube.DirectColor.24.reds: 7
DPSColorCube.DirectColor.24.greens: 7
DPSColorCube.DirectColor.24.blues: 7
DPSColorCube.DirectColor.24.grays: 7
```

If none of the above defaults apply to the display, the Client Library uses no color cube and a 2-element gray ramp; that is, black and white.

SEE ALSO `pswrap(1)`, `dpsexec(6)`

NOTES Copyright 1988-1992 Adobe Systems Incorporated.
PostScript and Display PostScript are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.

| | |
|--------------------|--|
| NAME | dpsexec – Display PostScript Executive |
| SYNOPSIS | dpsexec [-display <i>name</i>] [-sync] [-backup] [-noexec] |
| DESCRIPTION | <p>dpsexec is a Display PostScript program that allows the user to interact directly with the PostScript interpreter through a command interface. dpsexec reads lines of text from standard input and passes each line to the PostScript interpreter for execution. It creates a window that displays the results of graphics operations as they are executed. dpsexec exits when end of file is reached on standard input, or when the user types "quit<return>", which executes the PostScript quit operator.</p> <p>By default, dpsexec executes the PostScript executive operator before it accepts any user input. This operator puts the PostScript interpreter in "interactive executive" mode so that the user can control the interpreter directly. In this mode, the PostScript interpreter supports certain line-editing functions and prompts the user when it is ready to execute more input. See section 2.4.4, "Using the Interpreter Interactively," of the <i>PostScript Language Reference Manual, Second Edition</i>, for detailed information on this mode of operation.</p> |
| OPTIONS | <p>-display <i>name</i> Specifies the display on which to open a connection to the Display PostScript system. If no display is specified, the DISPLAY environment variable is used.</p> <p>-sync Establishes a synchronous connection with the specified X display.</p> <p>-backup Uses backing store for the window in which graphics are displayed, if possible.</p> <p>-noexec Prevents dpsexec from entering "interactive executive" mode. The primary effect of this option is to inhibit printing the PS> prompt before each line of input is accepted. This option is useful when dpsexec is run with standard input redirected from a file or a pipe.</p> |
| DIAGNOSTICS | PostScript language error messages are printed to standard output. |
| AUTHOR | Adobe Systems Incorporated |
| NOTES | <p>PostScript and Display PostScript are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.</p> <p>Copyright (c) 1990-1991 Adobe Systems Incorporated. All rights reserved.</p> |

| | |
|--------------------|---|
| NAME | dsdm – Drop Site Database Manager |
| SYNOPSIS | dsdm [-x] |
| DESCRIPTION | <p>The dsdm manages a database of all drop sites on the screen. When a drag-and-drop operation is started, the dsdm is queried for a list of the drop sites. This drop site information is used by the dragging application to provide user feedback during the drag operation.</p> <p>The dsdm is not used during normal operation of the system. Drop site database management is normally provided by the olwm(1) window manager. If you are running a window manager other than olwm, you must run dsdm in order for proper feedback to appear during drag-and-drop operations. The dsdm is normally started from the openwin-sys file.</p> <p>There should be only one dsdm running at a time. The dsdm will refuse to run if there is another dsdm running, or if olwm is running and is managing the drop site database. This mutual exclusion is accomplished through ownership of an X selection called _SUN_DRAGDROP_DSDM. If dsdm loses ownership of this selection, it will exit.</p> |
| OPTIONS | <p>-x Sets the owner of the _SUN_DRAGDROP_DSDM to None. If another instance of the dsdm is running, it will exit. If olwm is running and is maintaining the drop site database, it will stop maintaining the database.</p> |
| DIAGNOSTICS | <p>another DSDM is already running Another client has ownership of the _SUN_DRAGDROP_DSDM selection.</p> |
| SEE ALSO | olwm (1) |

| | |
|---------------------|--|
| NAME | ds_server_init – store a property in the X11 server to reduce colormap flashing within the DeskSet applications |
| SYNOPSIS | ds_server_init [<i>-f filename</i>] [<i>-a</i>] |
| AVAILABILITY | Available with the OpenWindows environment. |
| DESCRIPTION | <p>ds_server_init is a utility which stores a property on the server representing the set of colors to be designated as the DeskSet colors. This utility reduces colormap flashing within applications requiring colors by restricting applications to allocate colors only from the DeskSet colors. As an application requires a color, a best fit color algorithm is performed upon the DeskSet colors to return the closest match from the DeskSet color list, thus avoiding to allocate duplicate colors that vary slightly.</p> <p>The property is stored as a character string under the name <code>_SUN_DESKSET_COLORS</code> where the property string consists of a series of color strings in the hexadecimal form of color specification. The DeskSet colors are read in from the configurable file <code>\$OPENWINHOME/share/xnews/client/ds_server_init/ds_colors.txt</code>. If this file does not exist, ds_server_init will default to a set of its own back-up colors to store in the property. ds_server_init will allow up to 240 colors to be set on the property.</p> <p>There are four additional colors (if they exist) that are added to the DeskSet color list besides the colors specified in the color file. These colors are stored in the user's <code>\$HOME/.Xdefaults</code> file and are 1) <code>Window.ForegroundColor</code>, 2) <code>Window.BackgroundColor</code>, 3) <code>OpenWindows.WorkspaceColor</code>, and 4) <code>OpenWindows.WindowColor</code>.</p> <p>Currently the only applications using this property are the DeskSet tools, but this concept can work on any application wishing to share colors with the tools.</p> |
| INSTALLATION | ds_server_init is invoked when the X11 server is started. |
| OPTIONS | <p><i>-f filename</i> Indicates the filename containing the colors. <code>\$OPENWINHOME/share/xnews/client/ds_server_init/ds_colors.txt</code> is the default file.</p> <p><i>-a</i> This will force the allocation of the DeskSet colors at the time ds_server_init is invoked. By default, ds_server_init will allocate each color on an as-needed basis controlled from within the applications.</p> |
| FILES | <p><code>\$OPENWINHOME/share/xnews/client/ds_server_init/ds_colors.txt</code> This file contains the set of DeskSet colors in ASCII where the ds_server_init utility reads the colors from.</p> <p><code>\$HOME/.Xdefaults</code></p> |

| | | | | | | | | | |
|-------------------------|--|----------|--|--------|--|--------------|---|-------------------------|--|
| NAME | editres – a dynamic resource editor for X Toolkit applications | | | | | | | | |
| SYNTAX | editres [<i>-toolkitoptions...</i>] | | | | | | | | |
| OPTIONS | Editres accepts all of the standard X Toolkit command line options (see X11(7)). The order of the command line options is not important. | | | | | | | | |
| DESCRIPTION | Editres is a tool that allows users and application developers to view 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. Once the user is happy with a resource specification editres will append the resource string to the user's X Resources file. | | | | | | | | |
| USING EDITRES | <p>Editres provides a window consisting of the following four areas:</p> <table border="0"> <tr> <td style="padding-right: 20px;">Menu Bar</td> <td>A set of popup menus that allow you full access to editres's features.</td> </tr> <tr> <td>Panner</td> <td>The panner allows a more intuitive way to scroll the application tree display.</td> </tr> <tr> <td>Message Area</td> <td>Displays information to the user about the action that editres expects of the her.</td> </tr> <tr> <td>Application Widget Tree</td> <td>This area will be used to display the selected client's widget tree.</td> </tr> </table> <p>To begin an editres session select the Get Widget Tree menu item from the command menu. This will change the pointer cursor to cross hair. You should now select the application you wish look at by clicking on any of its windows. If this application understands the editres protocol then editres will display the client's widget tree in its tree window. If the application does not understand the editres protocol editres will inform you of this fact in the message area after a few seconds delay.</p> <p>Once you have a widget tree you may now select any of the other menu options. The effect of each of these is described below.</p> | Menu Bar | A set of popup menus that allow you full access to editres 's features. | Panner | The panner allows a more intuitive way to scroll the application tree display. | Message Area | Displays information to the user about the action that editres expects of the her. | Application Widget Tree | This area will be used to display the selected client's widget tree. |
| Menu Bar | A set of popup menus that allow you full access to editres 's features. | | | | | | | | |
| Panner | The panner allows a more intuitive way to scroll the application tree display. | | | | | | | | |
| Message Area | Displays information to the user about the action that editres expects of the her. | | | | | | | | |
| Application Widget Tree | This area will be used to display the selected client's widget tree. | | | | | | | | |
| COMMANDS | <p>Send Widget Tree Allows the user to click on any client that speaks the editres protocol and receive its widget tree.</p> <p>Refresh Widget Tree Editres only knows about the widgets that exist at the present time. Many applications create and destroy widgets "on-the-fly". Selecting this menu item will cause editres to ask the application to resend its widget tree, thus updating its information to the new state of the application.</p> <p>Example Xman only creates the widgets for its <i>topbox</i> when it starts up. None of the widgets for the manual page window are created until the user actually clicks on the <i>Manual Page</i> button. If you retrieved xman's widget tree before the the</p> | | | | | | | | |

manual page is active, you may wish to refresh the widget tree after the manual page has been displayed. This will allow you to also edit the manual page's resources.

Select Widget in Client

This menu item allows you to select any widget in the application, editres will then highlight the corresponding element the widget tree display. Once this menu item is selected the pointer cursor will again turn to a crosshair, and you must click any pointer button in the widget you wish to have displayed. Since some widgets are fully obscured by their children, it is not possible to get to every widget this way, but this mechanism does give very useful feedback between the elements in the widget tree and those in the actual client.

Dump Widget Tree to a File

For documenting applications it is often useful to be able to dump the entire application widget tree to an ascii file. This file can then be included in the manual page. When this menu item is selected a popup dialog is activated. Type the name of the file in this dialog, and either select *okay*, or type a carriage-return. Editres will now dump the widget tree to this file. To cancel the file dialog just select the *cancel* button.

Show Active Widgets

This command is the inverse of the **Select Widget in Client** command, it will show the user each widget that is currently selected in the widget tree, by flashing the corresponding widget in the application *numFlashes* (three by default) times in the *flashColor*.

Show Resource Box

This command will popup a resource box for the current client. This resource box (described in detail below) will allow the user to see exactly which resources can be set for the widget that is currently selected in the widget tree display. Only one widget may be currently selected, if greater or fewer are selected editres will refuse to pop up the resource box, and put an error message in the **Message Area**.

Show Author

Pops up message telling you who wrote this application.

Quit Exits editres.

TREE COMMANDS

The **Tree** menu contains several commands that allow operations to be performed on the widget tree.

Select All

Unselect All

Invert All

These functions allow the user to select, unselect, or invert all widgets in the widget tree.

Select Children

Select Parents

These functions select the immediate parent or children of each of the currently selected widgets.

Select Descendants

Select Ancestors

These functions select all parents or children of each of the currently selected widgets. This is a recursive search.

Show Widget Names

Show Class Names

Show Widget Windows

When the tree widget is initially displayed the labels of each widget in the tree correspond to the widget names. These functions will cause the label of **all** widgets in the tree to be changed to show the class name, IDs, or window associated with each widget in the application. The widget IDs, and windows are shown as hex numbers.

In addition there are keyboard accelerators for each of the Tree operations. If the input focus is over an individual widget in the tree, then that operation will only effect that widget. If the input focus is in the Tree background it will have exactly the same effect as the corresponding menu item.

The translation entries shown may be applied to any widget in the application. If that widget is a child of the Tree widget, then it will only affect that widget, otherwise it will have the same effect as the commands in the tree menu.

| Key | Option | Translation Entry |
|-------|---------------------|---------------------|
| space | Unselect | Select(nothing) |
| s | Select | Select(all) |
| i | Invert | Select(invert) |
| c | Select Children | Select(children) |
| d | Select Descendants | Select(descendants) |
| p | Select Parent | Select(parent) |
| a | Select Ancestors | Select(ancestors) |
| N | Show Widget Names | Relabel(name) |
| C | Show Class Names | Relabel(class) |
| I | Show Widget IDs | Relabel(id) |
| W | Show Widget Windows | Relabel(window) |

USING THE RESOURCE BOX

The resource box contains five different areas. Each of the areas, as they appear on the screen, from top to bottom will be discussed.

The Resource Line

This area at the top of the resource box shows the current resource entry exactly as it would appear if you were to save it to a file.

The Widget Names and Classes

This area allows you to select exactly which widgets this resource will apply to.

The area contains three lines, the first contains the name of the selected widget and all its ancestors, and the more restrictive dot (.) separator. The second line contains less specific the Class names of each widget, and well as the less restrictive star (*) separator. The last line contains a set of special buttons called **Allow Any Widget** which will completely compress out this level of the widget hierarchy, replacing it with just the star separator.

The initial state of this area is the most restrictive, using the resource names and the dot separator. By selecting the other buttons in this area you can ease the restrictions to allow more and more widgets to match the specification. The extreme case is to select all the **Allow Any Widget** buttons, which will match every widget in the application. As you select different buttons the tree display will update to show you exactly which widgets will be effected by the current resource specification.

Normal and Constrain Resources

The next area allows you to select the name of the normal or constraint resources you wish to set. Some widgets may not have constraint resources, so that area will not appear.

Resource Value

This next area allows you to enter the resource value. This value should be entered exactly as you would type a line into your resource file. Thus it should contain no unescaped new-lines. There are a few special character sequences for this file:

\n - This will be replaced with a newline.

\### - Where # is any octal digit. This will be replaced with a single byte that contains this sequence interpreted as an octal number. For example, a value containing a NULL byte can be stored by specifying \000.

\<new-line> - This will compress to nothing.

\\ - This will compress to a single backslash.

Command Area

This area contains several command buttons that I will describe in this section.

Set Save File

This button allows the user to modify file that the resources will be saved to. This button will bring up a dialog box that will ask you for a filename, once the filename has been entered, either hit carriage-return or click on the *okay* button. To popdown the dialog box without changing the save file, click the *cancel* button.

Save This button will append the **resource line** described above to the end of the current save file. If no save file has been set the **Set Save File** dialog box will be popped up to prompt the user for a filename.

Apply This button attempts to perform a XtSetValues call on all widgets that match the

resource line described above. The value specified is applied directly to all matching widgets. This behavior is an attempt to give a dynamic feel to the resource editor. Since this feature allows users to put an application in states it may not be willing to handle, a hook has been provided to allow specific clients block these SetValues requests (see **Blocking Editres Requests** below).

Unfortunately due to design constraints imposed on the widgets by the X Toolkit and the Resource Manager, trying to coerce an inherently static system into dynamic behavior can cause strange results. There is no guarantee that the results of an apply will be the same as what will happen when you save the value, and restart the application. This functionality is provided to try to give you a rough feel for what your changes will accomplish, and the results obtained should be considered suspect at best. Having said that, this is one of the neatest features of editres, and I strongly suggest that you play with it, and see what it can do.

Save and Apply

This button combines the Save and Apply actions described above into one button.

Popdown Resource Box

This button will remove the resource box from the display.

BLOCKING EDITRES REQUESTS

The editres protocol has been built into the Athena Widget set. This allows all application that are linked against Xaw to be able to speak to the resource editor. While this provides great flexibility, and is a useful tool, it can quite easily be abused. It is therefore possible for any Xaw client to specify a value for the **editresBlock** resource described below, to keep editres from divulging information about its internals, or to disable the **SetValues** part of the protocol.

editresBlock (Class **EditresBlock**)

Specifies which type of blocking this client wishes to impose on the editres protocol.

The accepted values are:

- | | |
|-----------|--|
| all | Block all requests. |
| setValues | Block all setvalues request, this is the only editres request that actually modifies the application, this is in effect stating that the application is read-only. |
| none | Allow all editres requests. |

Remember that these resources are set on any Xaw client, **not editres**. They allow individual clients to keep all or some of the requests editres makes from ever succeeding. Of course, editres is also an Xaw client, so it may also be viewed and modified by editres (rather recursive, I know), these commands can be blocked by setting the **editresBlock** resource on editres itself.

RESOURCES

For **editres** the available application resources are:

numFlashes (Class **NumFlashes**)

Specifies the number of times the widgets in the client application will be flashed when the **Show Active Widgets** command is invoked.

flashTime (Class **FlashTime**)

Amount of time between the flashes described above.

flashColor (Class **flashColor**)

Specifies the color used to flash client widgets. A bright color should be used that will immediately draw your attention to the area being flashed, such as red or yellow.

saveResourcesFile (Class **SaveResourcesFile**)

This is the file the resource line will be appended to when the **Save** button is activated in the resource box.

WIDGETS

In order to specify resources, it is useful to know the hierarchy of the widgets which compose **editres**. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

Editres editres

 Paned paned

 Box box

 MenuButton commands

 SimpleMenu menu

 SmeBSB sendTree

 SmeBSB refreshTree

 SmeBSB showClientWidget

 SmeLine line

 SmeBSB dumpTreeToFile

 SmeBSB flashActiveWidgets

 SmeBSB getResourceList

 SmeLine line

 SmeBSB quit

 MenuButton treeCommands

 SimpleMenu menu

 SmeBSB selectAll

 SmeBSB unselectAll

 SmeBSB invertAll

 SmeLine line

 SmeBSB selectChildren

 SmeBSB selectParent

 SmeBSB selectDescendants

 SmeBSB selectAncestors

 SmeLine line

 SmeBSB showWidgetNames

```

SmeBSB showClassNames
SmeBSB showWidgetIDs
SmeBSB showWidgetWindows
Paned hPane
  Panner panner
  Label userMessage
  Grip grip
Porthole porthole
Tree tree
  Toggle <name of widget in client>
  .
  .
  .
  TransientShell resourceBox
  Paned pane
  Label resourceLabel
  Form namesAndClasses
  Toggle dot
  Toggle star
  Toggle any
  Toggle name
  Toggle class
  .
  .
  .
  Label namesLabel
  List namesList
  Label constraintLabel
  List constraintList
  Form valueForm
  Label valueLabel
  Text valueText
  Box commandBox
  Command setFile
  Command save
  Command apply
  Command saveAndApply
  Command cancel
  Grip grip
Grip grip

```

ENVIRONMENT**DISPLAY**

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

FILES /usr/openwin/lib/app-defaults/Editres - specifies required resources

SEE ALSO X11(7), xrdb(1), Athena Widget Set

COPYRIGHT Copyright 1990, Massachusetts Institute of Technology.
See X11(7) for a full statement of rights and permissions.

AUTHOR Chris D. Peterson, formerly MIT X Consortium

| | |
|--------------------|---|
| NAME | edit_demo – ToolTalk client/server demonstration programs |
| SYNOPSIS | cntl edit |
| DESCRIPTION | cntl and edit are two related demo programs provided with the ToolTalk product. Both programs are compiled by running the make(1S) command in the directory <code>\$OPENWINHOME/share/src/tooltalk/demo/edit_demo</code> . Your <code>OPENWINHOME</code> environment variable must be set to where OpenWindows is installed (typically <code>/usr/openwin</code>). After compiling, run the cntl (for 'control') program. This will popup an OpenWindows application with a single button and a text input field. Enter a filename in the text field and load it for editing by single clicking on the button. This will use ToolTalk to invoke the edit program as another OpenWindows popup application. From there you can select text in the edit window, and click on the Make Object button on the panel of the edit popup. Then using the cntl program button you can use ToolTalk to highlight the object you just made in the other window. You can use ttsnoop to monitor the contents of the ToolTalk messages sent by edit . |
| SEE ALSO | ttsnoop(6) ttsession(1) |
| DIAGNOSTICS | If you try and invoke cntl (or any ToolTalk application) and you get a message saying the application could not start ToolTalk, or ttsession , make sure that you have one of the environment variables <code>DISPLAY</code> or <code>_SUN_TT_SESSION</code> set, and that ttsession is in your <code>PATH</code> , or that the <code>SUN_TTSESSION_CMD</code> environment variable indicates where the ttsession program resides. For more information on ttsession and the environment variables it uses, see the ttsession man page. |

| | |
|--------------------|---|
| NAME | filemgr – a GUI-based file management application |
| SYNOPSIS | filemgr [-C] [-a] [-c] [-d <i>directory</i>] [-i <i>secs</i>] [-M] [-name <i>app-name</i>] [-r] [-v] [-?] |
| DESCRIPTION | filemgr is a file management program that lets you navigate through directories and manipulate files. filemgr lets you find, create, copy, move, link, open, and print files using a graphical user interface (GUI). You can also change file properties, create new directories, and view multiple directories at the same time. |
| OPTIONS | <p>In addition to the generic tool arguments described in xview(7), filemgr can accept the following options:</p> <ul style="list-style-type: none"> -a This will force filemgr to check both folder and file modification times. By default, filemgr only checks the folder modification times. Be careful, as this option will severely affect performance. -C This will start filemgr without using the Classing Engine to try to determine what each file type is. This means the only three types of file icons will be displayed: the generic document, folder and application icons. -c This will force filemgr to display file pane items by columns, rather than by rows. -d <i>directory</i> This will start filemgr in the given <i>directory</i>. -i <i>secs</i> This will set the timer to check on folder and/or file modification times every <i>secs</i> seconds. -M Just display using the foreground and background colors only, even on a color screen. -name <i>app-name</i> This option instructs filemgr to use resources associated with <i>app-name</i> in your \$HOME/.desksetdefaults file instead of using resources for filemgr. This lets you run multiple filemgrs at the same time with different attributes. <i>app-name</i> should not contain "." or "*" characters. -r This will force filemgr to display file pane items by rows, rather than by columns (default). -v, -ver, -version Any of these options will display the current version number of filemgr. -? This will display a partial list of command line options which can be passed to filemgr. For additional generic options which can be passed to filemgr, type "man xview -?". Note that csh(1) shell users will have to enter a backslash ("\") in front of the "?" to avoid having their shell expand the regular expression. |
| RESOURCES | On startup, filemgr will use the following X resources which are stored in \$HOME/.desksetdefaults . Note these resource names will be prepended with deskset.filemgr . |

| | |
|---------------------|--|
| Resource: | applicationColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the default application icon. |
| Resource: | autoShowCD |
| Values: | True, False (True) |
| Description: | If set to true, when a CD is inserted, then filemgr will display it's contents in a new window. |
| Resource: | autoSortOnUpdate |
| Values: | True, False (False) |
| Description: | If set to true, when a file is added or deleted in a directory being viewed by filemgr, the contents are automatically resorted (based on the current sort type). This means it is impossible to retain any positional information. This is similar to the way that previous versions of filemgr worked and may be the preference of some users. |
| Resource: | cacheSize |
| Values: | 0-100000000 (1500000) |
| Description: | Determines the limit (in bytes) for the garbage collection run by the fmgc(1) program (started by filemgr). fmgc will continue to prune the ~/fm hierarchy until this limit is reached. |
| Resource: | cdromContentMatch |
| Values: | True, False (True) |
| Description: | Determines whether the Classing Engine will do matching by contents for CDROM files. Matching by contents is a slow operation because of the speed of CDROM devices. |
| Resource: | classingEngineBufferSize |
| Values: | 0-32768 (512) |
| Description: | Denotes how many bytes are read from each file, and passed onto the Classing Engine to try to determine it's file type. |
| Resource: | confirmDelete |
| Values: | True, False (True) |
| Description: | When true, sends deleted files to the wastebasket. Otherwise, files are deleted normally as if you had run a rm(1) command. |
| Resource: | confirmDeleteFolder |
| Values: | True, False (True) |
| Description: | When true, the user will be prompted whenever they try to delete a folder (and therefore it's contents) to the Wastebasket. Setting this resource to false disables this prompt. |

| | |
|---------------------|---|
| Resource: | confirmDestroyFolder |
| Values: | True, False (True) |
| Description: | When true, the user will be prompted whenever they try to destroy a folder (and therefore it's contents). This could be either the Destroy option from the Wastebasket, or if they have set their Edit menu item to destroy (from the general property panel). Setting this resource to false disables this prompt. |
| Resource: | confirmQuit |
| Values: | True, False (True) |
| Description: | When true, the user will be prompted when they select the "Quit File Manager" option from the File button menu, to ask if they really want to do this. Setting this resource to false disables this prompt. |
| Resource: | documentColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the default application icon. |
| Resource: | fileCheckInterval |
| Values: | 0-9999 (5) |
| Description: | The amount of time (in seconds) between checks on the various monitored file systems to see if changes have occurred. |
| Resource: | filenameNoChars |
| Values: | 0-255 (255) |
| Description: | The maximum number of characters to display in a filename. Note this is based on the average width of characters, so when using proportional fonts the number of characters displayed will vary. |
| Resource: | filterScript |
| Values: | regular expression (NULL) |
| Description: | A regular expression describing which files you wish to view. The default is empty, which indicates all files should be displayed. |
| Resource: | floppyContentMatch |
| Values: | True, False (True) |
| Description: | Determines whether the Classing Engine will do matching by contents for floppy disk files. Matching by contents is a slow operation because of the speed of floppy disk devices. |
| Resource: | folderColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the default folder icon. |
| Resource: | followSymbolicLinks |

| | |
|---------------------|---|
| Values: | True, False (False) |
| Description: | When false, filemgr doesn't follow symbolic links when changing directories. Note that prior to Solaris 2.3, the automounter used a symbolic link when automounting directories. Setting this resource to true will not correctly monitor such automounted directories. |
| Resource: | iconDirectionVertical |
| Values: | True, False (False) |
| Description: | When false, displays files row-by-column. Otherwise, displays files column-by-row. |
| Resource: | iconListStyle |
| Values: | int (0) |
| Description: | Saves the state of the "List Options" toggles on the Customize View popup. This is not very human readable and should be changed in the future. |
| Resource: | loadDirectoryState |
| Values: | True, False (True) |
| Description: | When false, directory state information for each directory will not be read from or written to the users <code>~/.fm</code> directory. |
| Resource: | loadIconPositions |
| Values: | True, False (True) |
| Description: | When false, icon positional information for each directory will not be read from or written to the users <code>~/.fm</code> directory. Setting this resource to true, overrides the <code>loadDirectoryState</code> resource. |
| Resource: | maxGotoMenuEntries |
| Values: | 1-500 (10) |
| Description: | The maximum number of entries that will be displayed in the Goto button menu. |
| Resource: | newFolderName |
| Values: | String (NewFolder) |
| Description: | The initial name given to newly created folders. |
| Resource: | newWindowOnDirectoryOpen |
| Values: | True, False (False) |
| Description: | Whether a new popup sub-folder pane is created when a folder is opened. |
| Resource: | otherEditor |
| Values: | UNIX command (<code>shelltool sh -c "sleep 3 ; vi \$FILE"</code>) |
| Description: | The open method the user wishes to use for opening files. Note this |

should provide its own window or be started in a shelltool to work correctly. The sleep command is added so the window's dimensions will be set when the editor starts.

| | |
|---------------------|---|
| Resource: | printScript |
| Values: | UNIX command (<code>cat \$FILE mp -lo lp</code>) |
| Description: | The print method the user wishes to use for printing files. |
| Resource: | shellToolName |
| Values: | UNIX command (<code>cmdtool</code>) |
| Description: | The type of tty window to use when running the Custom Command's UNIX Shell menu option. |
| Resource: | showHidden |
| Values: | True, False (False) |
| Description: | When True, displays hidden files, i.e. dot files. |
| Resource: | sortCaseSensitive |
| Values: | True, False (False) |
| Description: | When True, sorting by name is a case sensitive operation with filenames in uppercase appearing first. |
| Resource: | sortType |
| Values: | Name, Type, Size, Date (Name) |
| Description: | Saves the state of the "Sort By" toggles on the Customize View popup. |
| Resource: | treeClosed |
| Values: | True, False (True) |
| Description: | When True, start the folder view window in iconic mode. |
| Resource: | treeDirectionVertical |
| Values: | True, False (False) |
| Description: | Determines the orientation of the folder view. |
| Resource: | treeHeight |
| Values: | 0-"Height of Screen" (18 lines) |
| Description: | Saves the height of the open folder view window (in pixels). |
| Resource: | treeIconXPosition |
| Values: | 0-"Width of Screen" (0) |
| Description: | Saves the X location of the folder view window icon (in pixels). |
| Resource: | treeIconYPosition |
| Values: | 0-"Height of Screen" (0) |
| Description: | Saves the Y location of the folder view window icon (in pixels). |

| | |
|--------------------|---|
| Resource: | treePaneGap |
| Values: | 48-320 (96) |
| Description | Gap between one level of icons in the folder view, and the next. Adjusting this value is a means of condensing the amount of space the folder view uses. |
| Resource: | treeView |
| Values: | True, False (False) |
| Description | When True, display the folder view window on startup. |
| Resource: | treeWidth |
| Values: | 0-"Width of Screen" (40 characters) |
| Description | Saves the width of the open folder view window (in pixels). |
| Resource: | treeWindowXPosition |
| Values: | 0-"Width of Screen" (0) |
| Description | Saves the X location of the open folder view window (in pixels). |
| Resource: | treeWindowYPosition |
| Values: | 0-"Height of Screen" (0) |
| Description | Saves the Y location of the open folder view window (in pixels). |
| Resource: | useCache |
| Values: | True, False (True) |
| Description | When False, filemgr will not read or write directory cache information into the users <code>~/.fm</code> hierarchy. |
| Resource: | useClassingEngine |
| Values: | True, False (True) |
| Description | When False, the Classing Engine is not used to try to determine what each file type is. This means the only three types of file icon will be displayed; the generic document, folder and application icons. |
| Resource: | useTextedit |
| Values: | True, False (True) |
| Description | When True, use the texteditor to open files. Otherwise, use the user defined open method described above under "otherEditor". |
| Resource: | viewType |
| Values: | Icon, List, Content (Icon) |
| Description | Saves the state of the "Display Mode" toggles on the Customize View popup. |
| Resource: | wastebasketClosed |
| Values: | True, False (True) |

| | |
|--------------------|---|
| Description | When True, start the wastebasket in iconic mode. |
| Resource: | wastebasketHeight |
| Values: | 0-"Height of Screen" (18 lines) |
| Description | Saves the height of the open wastebasket (in pixels). |
| Resource: | wastebasketIconXPosition |
| Values: | 0-"Width of Screen" (0) |
| Description | Saves the X location of the wastebasket icon (in pixels). |
| Resource: | wastebasketIconYPosition |
| Values: | 0-"Height of Screen" (0) |
| Description | Saves the Y location of the wastebasket icon (in pixels). |
| Resource: | wastebasketWidth |
| Values: | 0-"Width of Screen" (40 characters) |
| Description | Saves the width of the open wastebasket (in pixels). |
| Resource: | wastebasketWindowXPosition |
| Values: | 0-"Width of Screen" (0) |
| Description | Saves the X location of the open wastebasket (in pixels). |
| Resource: | wastebasketWindowYPosition |
| Values: | 0-"Height of Screen" (0) |
| Description | Saves the Y location of the open wastebasket (in pixels). |

USAGE

filemgr operates via a set of pulldown menus from button stacks in a control panel. Most menu commands operate on the currently selected files. To select a file, click the SELECT mouse button on its icon. To select additional files, click with the ADJUST mouse button on additional files. The menu commands are described below.

File >

Open...

Opening a document will open the document in the appropriate application. Opening a folder will create a subfolder showing the items in that folder. Opening an application starts the application. This is the default action of a double-clicked file.

Open in Editor...

Opens the selected items using the users preferred editor. See *File Manager Properties* (below).

Create Folder

Creates an empty subfolder in the file pane.

Duplicate

Create a duplicate of each of the selected files.

Print One

Prints the selected items using their default print method. See *File*

Manager Properties below.

- Print...** Displays a command window which allows you to print your selected files with a print method you specify.
- Find...** Displays a command window which allows you to search for files within the specified folders.
- Information...**
This command window allows you to see and change the file attributes of the selected files.
- Remote Copy...**
Displays a command window which allows you to copy files to and from remote machines. You can make a reference to files on a remote machine by using the format *machine_name:file_name*. See **rcp(1)**.
- Custom Commands**
Displays a menu containing custom commands you have created.
- UNIX Shell...**
Will bring up a command window where you can directly enter UNIX commands.
- Format Disk...**
Will display a command window allowing you to format a floppy diskette (in either DOS or Unix format).
- Rename Disk...**
Will display a command window allowing you to rename a floppy diskette.
- Comments...**
Displays a command window where you can send comments to the **filemgr** development team. These comments would be related to bugs, problems or inconsistencies found with the **filemgr** program.
- Quit File Manager**
Allows you to really quit the **filemgr** program and not just the current window.

View >

- Open Folder View**
Will show a hierarchical display of folders in a separate window.
- Large Icon View**
Displays file pane items as large icons in a positioned view.
- Small Icon View**
Displays file pane items as small icons in a positioned view.
- Icon by Name**
Displays file pane items as large icons sorted by name.
- Icon by Type**
Displays file pane items as large icons sorted by type and name.
- List by Name**
Displays file pane items as small icons, one per line, sorted by name.
- List by Type**
Displays file pane items as small icons, one per line, sorted by type and

name.

List by Size

Displays file pane items as small icons, one per line, sorted by size.

List by Date

Displays file pane items as small icons, one per line, sorted by date.

Cleanup Icons (or Selection)

Will move each file pane item to its nearest grid point. If one or more files are selected, then this operation only affects the selected files.

Edit >

Select All

Selects all files in the current folder.

Cut Places selected files on the clipboard for a pending move operation.

Copy Places selected files on the clipboard for a pending copy operation.

Link Places selected files on the clipboard for a pending link operation.

Paste Pastes files from clipboard into the current folder. Pasted files are linked, copied, or moved, depending on how they were first stored in the clipboard.

Delete (or Destroy)

Deletes selected files to the wastebasket. To have files destroyed without going to the wastebasket (and with no hope of recovery), select the "Destroy" option from the "Edit Menu item is" item on the "General Defaults" sheet in *File Manager Properties*.

Properties...

Displays a multi-level property sheet allowing you to customize the behavior of **filemgr**.

Goto > Used in conjunction with the destination entered on the goto line (located at the right of the goto button). To change directories, type the directory name on the goto line and then press *Goto*.

If nothing is entered on the Goto line, then the first folder displayed on the menu will be "Home", which will return you to your home folder. Next in the menu are two application specific entries for viewing the Folder View and the Waste Basket. If there are any floppies or CDs being viewed, they also have an entry in this section of the menu. Finally the menu displays the last folders you have visited. Selecting a folder causes that folder to be opened.

SEE ALSO

find(1), fmgc(1), egrep(1), xview(7), binder(1), rcp(1)

Solaris User's Guide

"About File Manager" in the Help Handbook available through the Help option on the Workspace menu.

FILES

\$HOME/.fmcmd

This file contains the old user defined custom commands, which can be referenced by the "Custom Commands" menu. When these commands are written out again, they will be written to the **\$HOME/.desksetdefaults** file.

\$HOME/.desksetdefaults

This file saves the state of the DeskSet tools in X resource format. Do not edit this file by hand as changes will be lost when the Server writes to this file!

BUGS

Comments/formatting in the **\$HOME/.desksetdefaults** file can be lost when the server saves the tool's state.

Choosing a custom command which requires input (such as **rm -i \$FILE**) can cause **filemgr** to hang. Custom commands needs to be more robust.

Print methods, open methods, and custom commands depend on the **\$FILE** variable being set properly. Failure to include **\$FILE** will cause the executed command to behave improperly.

| | |
|--------------------|---|
| NAME | fixframe – convert FrameMaker 2.0 PostScript files to conform to the PostScript structuring conventions |
| SYNOPSIS | fixframe < <i>frame_file</i> > <i>output_file</i> |
| DESCRIPTION | <p>fixframe converts FrameMaker 2.0 POSTSCRIPT files to follow the POSTSCRIPT Document Structuring Conventions described in Appendix G of the POSTSCRIPT Language Reference Manual. pageview will not work with unaltered FrameMaker 2.0 POSTSCRIPT files.</p> <p>fixframe is automatically run by pageview when a user attempts to view a FrameMaker 2.0 POSTSCRIPT file using pageview.</p> <p>Specifically, fixframe saves font definitions that occur between pages and inserts them after the %%Page: comment for each page.</p> |
| SEE ALSO | <p>pageview(1)</p> <p><i>POSTSCRIPT Language Reference Manual</i>, Adobe Systems Inc., Addison-Wesley</p> |
| TRADEMARK | POSTSCRIPT is a registered trademark of Adobe Systems Inc |

| | |
|--------------------|--|
| NAME | fixinterleaf – convert Interleaf PostScript files to conform to the PostScript structuring conventions |
| SYNOPSIS | fixinterleaf <i>interleaf_file</i> <i>output_file</i> |
| DESCRIPTION | <p>fixinterleaf converts Interleaf 2.0 POSTSCRIPT files to follow the POSTSCRIPT Document Structuring Conventions described in Appendix G of the POSTSCRIPT Language Reference Manual. pageview will not work with unaltered Interleaf POSTSCRIPT files.</p> <p>fixinterleaf is automatically run by pageview when a user attempts to view an Interleaf POSTSCRIPT file using pageview.</p> <p>Specifically, fixinterleaf saves glyph definitions that occur between pages and inserts them into the Setup section of the document (after the Prolog). It also inserts the proper %%Page comments between each page, and it properly numbers pages from back to front.</p> |
| SEE ALSO | <p>pageview(1)</p> <p><i>POSTSCRIPT Language Reference Manual</i>, Adobe Systems Inc., Addison-Wesley</p> |
| TRADEMARK | POSTSCRIPT is a registered trademark of Adobe Systems Inc |

| | |
|--------------------|--|
| NAME | fmgc – a program to garbage collect a users ~/.fm directory hierarchy |
| SYNOPSIS | fmgc [-l <i>limit</i>] [-u <i>username</i>] [-v] [-V] [-?] [<i>directory ...</i>] |
| DESCRIPTION | <p>fmgc is a program to garbage collect a users ~/.fm directory hierarchy. This hierarchy of files is created by the filemgr program to store information on directories visited by the user (including the position of each file icon if the user moves a file). It is possible that over time, the size of this file hierarchy could grow considerably. This program is used to help the user maintain this. It will remove directory information relating to the oldest used directories until a limit is reached.</p> <p>It uses information stored in the users ~/.desksetdefaults file to generate a list of directories to be excluded from this pruning. This list will contain the directories for the users saved window positions plus any directories on the users goto list. It is also possible to include additional directories on the fmgc command line.</p> |
| OPTIONS | <p>-l <i>limit</i> This is the limit (in bytes) for the garbage collection. fmgc will continue to prune the ~/.fm hierarchy until this limit is reached. If not given, then the limit is set to 1.5 megabytes.</p> <p>-u <i>username</i> The name of the user that will have their ~/.fm directory hierarchy garbage collected. If not given, then fmgc will use the programs effective uid.</p> <p>-v This will display the current version number of fmgc.</p> <p>-V Be verbose. Output messages stating what's happening.</p> <p>-? This will display a list of command line options which can be passed to fmgc.</p> |
| SEE ALSO | filemgr (1), |
| FILES | <p>\$HOME/.fm This directory contains directory state, icon positional and file status information saved by the filemgr program.</p> <p>\$HOME/.desksetdefaults This file contains the state of the DeskSet tools in X resource format. fmgc reads various X resources pertaining to the filemgr program from this file.</p> |

| | |
|--------------------|---|
| NAME | format_floppy – diskette formatting program used by FileManager |
| SYNOPSIS | format_floppy -d <i>device</i> -m <i>mnt_point</i> [-h] [-n <i>name</i>] [-p <i>popup_type</i>] [-r] [-x <i>x_pos</i>] [-y <i>y_pos</i>] |
| DESCRIPTION | <p>Format_floppy is used by filemgr(1) to format 3.5 in. high-density floppy disks. It is not intended to be used directly but rather is called by filemgr when a floppy disk can not be read and the user indicates that it should be formatted. This situation occurs if the user inserts a new disk into the drive or tries to use a disk that is unreadable for some reason. The user will be notified that the disk is unreadable and be given the options of ejecting the disk or formatting it. Filemgr and vold(1) must be running when the unreadable disk is inserted into the drive in order for format_floppy to be called.</p> <p>Note that DOS-formatted disks are automatically recognized by vold and filemgr will not try to reformat them.</p> |
| OPTIONS | <p>-d <i>device</i> The raw device name of the floppy disk must be specified. This information is in the second field of the file /tmp/removable/floppyN where <i>N</i> is the number of the floppy disk (usually 0).</p> <p>-h Causes format_floppy to display its "help" or usage message.</p> <p>-m <i>mnt_point</i> The mount-point of the floppy disk must be specified. This information is in the first field of the file /tmp/removable/floppyN, where <i>N</i> is the number of the floppy disk (usually 0).</p> <p>-n <i>name</i> Allows a name to be assigned to the newly-formatted floppy disk.</p> <p>-p <i>popup_type</i> Allows the type of popup menu to be specified where <i>popup_type</i> can be format, unformatted, or unlabeled. The default is format. Note that the wording in these popup menus is different.</p> <p>-r Allows the floppy disk to be renamed. If -r is specified, -n must be also.</p> <p>-x <i>x_pos</i> Used with the -p option to specify the x-coordinate position for the popup menu.</p> <p>-y <i>y_pos</i> Used with the -p option to specify the y-coordinate position for the popup menu.</p> |
| FILES | /tmp/removable/* |
| SEE ALSO | filemgr (1), vold (1) |

| | |
|--------------------|--|
| NAME | helpopen – utility for programmatically controlling helpviewer(1) |
| SYNOPSIS | helpopen <i>help-handbook-file</i> |
| DESCRIPTION | helpopen is a utility for programmatically controlling helpviewer(1) . helpopen sends a ToolTalk request to a running helpviewer to display the specified help handbook file. If no helpviewer is running, helpopen starts one. OpenWindows users do not typically need to use helpopen directly. It is used to implement the "more help" feature in OpenWindows spot help. |
| OPTIONS | helpopen accepts the following command line options: <i>-f help-handbook-file</i> Specify the name of the help handbook to be viewed. The file name must be relative to a directory in \$HELPPATH. See helpviewer(1) for more information. |
| DIAGNOSTICS | helpopen prints an error message on failure and returns non-zero exit status. |
| BUGS | helpviewer and helpopen do not share the same view of the file system, since they are separate processes and could be running on different machines. |

| | |
|--------------------|--|
| NAME | helpviewer – OpenWindows viewer for on-line help handbooks |
| SYNOPSIS | helpviewer <i>-f help-handbook-file</i> |
| DESCRIPTION | <p>helpviewer is an OpenWindows application for viewing and navigating on-line help handbooks.</p> <p>helpviewer lets you page through a help handbook, magnify or reduce pages, and follow hypertext links and table-of-contents entries within a handbook and to other handbooks. Because help handbooks are in PostScript format, they may contain high quality fonts and graphics.</p> <p>You should not start helpviewer directly. It is started automatically by OpenWindows when you click SELECT on the "More" button in a spot help window, or when you select the "Help..." item in the OpenWindows root menu.</p> |
| OPTIONS | <p>helpviewer accepts most of the generic tool arguments described in xview(7), as well as the following options:</p> <p><i>-f help-handbook-file</i></p> <p style="padding-left: 2em;">Specify the name of the help handbook to be viewed. This file name should be specified relative to a directory in SHELPPATH (see ENVIRONMENT below).</p> |
| USAGE | <p>Once a help handbook is displayed in the helpviewer window, you can view it as follows:</p> <p>Page Turning</p> <p style="padding-left: 2em;">Click SELECT on the left and right arrow buttons at the top of the helpviewer window to page back and forth through a handbook. The PgUp and PgDn keys on the keyboard perform the same functions. In addition, the Home and End keys let you page through a handbook one chapter at a time.</p> <p>Page History</p> <p style="padding-left: 2em;">helpviewer keeps track of each page visited. Click SELECT on the Go Back button until the window returns to the page you want to revisit, or use the Undo keyboard function.</p> <p>Hypertext Links</p> <p style="padding-left: 2em;">helpviewer has a simple hypertext mechanism. Hypertext links are displayed as rectangular outlines around words or graphics on a page. Double-clicking SELECT on a link causes helpviewer to display the handbook and page to which the link points. Handbooks may have links to other handbooks. Certain types of hypertext links can also initiate system processes. Double-clicking on these links will start up a program or shell script.</p> <p>Page Magnification</p> <p style="padding-left: 2em;">Magnify or reduce the size of the handbook by pulling the resize corners on the helpviewer window, or select the "View->Custom Magnification" menu item. Select the "View->Standard Magnification" menu item to reset the handbook to the standard size. For very high magnifications, select the "View->Partial Page"</p> |

menu item so that the **helpviewer** window still fits on the screen.

ENVIRONMENT**HELPPATH**

This environment variable must be set to the directories that contain help handbooks (e.g. \$OPENWINHOME/lib/locale:\$OPENWINHOME/lib/help). The **helpviewer** uses the **\$HELPPATH** variable to locate handbook files. **\$HELPPATH** is typically set automatically during the OpenWindows startup process, so it is not usually necessary to set it yourself.

SEE ALSO

helpopen(1)

DIAGNOSTICS

helpviewer(1) displays PostScript error messages in the console.

NOTES

PostScript is a trademark of Adobe Systems Incorporated.

| | |
|--------------------|---|
| NAME | ico – animate an icosahedron |
| SYNOPSIS | ico [<i>display list</i>] [-geometry <i>geom</i>] [-d <i>pattern</i>] [-i] |
| DESCRIPTION | Ico displays a wire-frame rotating polyhedron, with hidden lines removed, or a solid-fill polyhedron with hidden faces removed. There are a number of different polyhedra available; adding a new polyhedron to the program is quite simple. |
| OPTIONS | -geometry <i>geom</i> Specify the size and/or location of the window. -d <i>pattern</i> Specify a bit pattern for drawing dashed lines for wire frames. -i Use inverted colors for wire frames. For each display specified, ico creates another window. The total geometry is split vertically amongst these windows, and the ico bounces between them. The end result is that ico foo:0 bar:0 would result in ico bouncing half on display foo:0 and half on bar:0 . |
| SEE ALSO | X11(7) , ico2(6) |
| BUGS | Doesn't deal too well with being resized. |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |

| | |
|-------------------------|--|
| NAME | ico2 – animate an icosahedron or other polyhedron |
| SYNOPSIS | ico2 [-r] [-d pattern] [-i] [-dbl] [-faces] [-noedges] [-sleep n] [-obj object] [-objhelp] [-colors color-list] |
| DESCRIPTION | Ico2 displays a wire-frame rotating polyhedron, with hidden lines removed, or a solid-fill polyhedron with hidden faces removed. There are a number of different polyhedra available; adding a new polyhedron to the program is quite simple. |
| OPTIONS | <p>-r Display on the root window instead of creating a new window.</p> <p>-d pattern Specify a bit pattern for drawing dashed lines for wire frames.</p> <p>-i Use inverted colors for wire frames.</p> <p>-dbl Use double buffering on the display. This works for either wire frame or solid fill drawings. For solid fill drawings, using this switch results in substantially smoother movement. Note that this requires twice as many bit planes as without double buffering. Since some colors are typically allocated by other programs, most eight-bit-plane displays will probably be limited to eight colors when using double buffering.</p> <p>-faces Draw filled faces instead of wire frames.</p> <p>-noedges Don't draw the wire frames. Typically used only when -faces is used.</p> <p>-sleep n Sleep <i>n</i> seconds between each move of the object.</p> <p>-obj object Specify what object to draw. If no object is specified, an icosahedron is drawn.</p> <p>-objhelp Print out a list of the available objects, along with information about each object.</p> <p>-colors colorlist Specify what colors should be used to draw the filled faces of the object. If less colors than faces are given, the colors are reused.</p> |
| ADDING POLYHEDRA | <p>If you have the source to ico, it is very easy to add more polyhedra. Each polyhedron is defined in an include file by the name of <i>objname.h</i>, where <i>name</i> is something related to the name of the polyhedron. The format of the include file is defined in the file <i>polyinfo.h</i>. Look at the file <i>objcube.h</i> to see what the exact format of an <i>objname.h</i> file should be, then create your <i>objname.h</i> file in that format.</p> <p>After making the new <i>objname.h</i> file (or copying in a new one from elsewhere), simply do a 'make depend'. This will recreate the file <i>allobjs.h</i>, which lists all of the <i>objname.h</i> files. Doing a 'make' after this will rebuild ico with the new object information.</p> |

BUGS

If *-dbl* or *-colors* is specified on a monochrome screen, or if the number of bit planes needed is more than what is available, the server hangs (this is on a Sun 3/60, patches up to 80 applied).

A separate color cell is allocated for each name in the *-colors* list, even when the same name may be specified twice.

SEE ALSO

ico(6)

| | |
|---------------------|--|
| NAME | iconedit – create and edit images for OpenWindows icons, cursors, and panel items |
| SYNOPSIS | iconedit [<i>filename</i>] [<i>XView command-line arguments</i>] |
| AVAILABILITY | This command is available with the OpenWindows user environment. For information about XView command-line arguments see the XView documentation. |
| OPTIONS | <i>filename</i> Contains the image. <i>XView command-line arguments</i> iconedit accepts XView command-line arguments listed in xview(7) . |
| DESCRIPTION | iconedit is part of the OpenWindows DeskSet. With iconedit you can create and edit small images for use in icons, cursors, panel items, etc. |
| | Main Window |
| | This window contains the controls for manipulating your image, as well as the image itself as it will appear in an icon or cursor, and an expanded image in a drawing canvas. From the main window you have the following pull-down menus. |
| | File |
| | Load Specify a file to load. |
| | Save Save the file under the current name. |
| | Save As Save file under a new name. |
| | Print Generate PostScript output for a printer or a file. |
| | View |
| | Grid On/Off |
| | Turns a grid on or off in the display area. One grid square is a 4x4 area of pixels. |
| | Edit |
| | Undo The last action is un-done. A record of undos available is kept in the lower right footer. |
| | Redo The last undo is redone, provided the last action was an undo. |
| | Clear The drawing area is cleared by having all of the pixels set to white. |
| | Cut A selected area is cleared, with its contents moved to the paste buffer. If no area is selected, the default area is the entire image. |
| | Copy A selected area is copied to the paste buffer. If no area is selected, the default area is the entire image. |
| | Paste The contents of the paste buffer are copied into the image at the current pointer location. |
| | Invert The selected area is inverted, black for white and white for black. This action is only available in B&W mode. |

Properties

Format Selects the default save format of an image, be it an XView icon, an X Bitmap, a color X Pixmap, or a monochrome X Pixmap.

Size Selects the size of the image. 64x64 is the most common size for an icon, 32x32 is used within **filemgr** and 16x16 is the size of cursors.

Palette This brings up the color palette for choosing the pen color.

Controls**Drawing Mode**

This choice lets you select your drawing mode: dots, lines, rectangles, circles, ellipses, text, selection, or erase.

Fill Pattern

This selects the pattern in which objects drawn are filled. The choices range from outline through various textures to solid. The fill pattern is only in effect when drawing rectangles, circles or ellipses.

Color or B&W

This item selects between having your image in color or monochrome. If you change it from color to monochrome, the color information is lost, though it can be restored through undo.

Movement Arrows

These let you move your image or a selected part of your image. Vertical and horizontal flip, as well as rotation are also provided.

SEE ALSO

xview(7)

Solaris User's Guide

XView documentation

"About Icon Editor" in the Help Handbook available through the Help option on the Workspace menu.

| | |
|--------------------|---|
| NAME | imagetool – Image viewer for OpenWindows |
| SYNOPSIS | imagetool [-usage] [-v] [-verbose] [-timeout seconds] [<i>imagefile</i>] |
| DESCRIPTION | Imagetool is an interactive image viewer. Imagetool can be used to view the contents of a variety of file types such as gif, tiff, jfif (jpeg) and POSTSCRIPT. The user may perform various operations on the image such as rotation, zooming and flipping to view the image differently. If the file loaded in is a multipage document (such as a POSTSCRIPT document), the user may page through the entire document, or skip to any page directly. |
| USAGE | The four menu buttons across the top of the main window are described below: <ul style="list-style-type: none"> File The File menu contains the following items. <ul style="list-style-type: none"> Open... Brings up a dialog which allows the user to select a file that is to be opened for viewing. Open As... Brings up a dialog which allows the user to select a file that is to be opened for viewing, and also allows the user to specify the file type. Save... If the file has no name, this brings up a dialog for the user to specify the file name, and save the file. If the file is named, then this simply saves the currently viewed image to the file. Save As... Brings up a dialog which allows the user to specify the file name, the file type, number of colors and compression type of the file that is to be saved. Save Selection As... Brings up a dialog which allows the user to specify the file name, the file type, number of colors and compression type of the file that is to be saved. This menu item is only active if the user has selected a region of interest in the currently displayed image. Save Page As Image... Brings up a dialog which allows the user to specify the file name, the file type, number of colors and compression type of the file that is to be saved. This menu item is only active if the user is viewing a page of a multipage file. Print One Allows user to print one copy of the currently displayed image using the current values (which may be the default values) on the Print dialog. Print Preview... Brings up a window and displays the image as it would look if it were printed, based on the settings (which may be the default values) on the Print dialog. Print... Brings up a dialog which allows user to set various parameters |

that affect the printing of the currently displayed image such as position, and size.

View

The View menu contains the following items.

Image Info...

Brings up a window that displays various information about the currently viewed image, such as width and height.

Page Overview...

Brings up a dialog which displays at most 16 pages of the currently viewed document. The user may select a page for viewing from this display. This option is valid only if the user is viewing a multipage document.

Page Viewing Controls

Brings up a dialog from which the user may set various parameters that affect the viewing of multipage (such as POSTSCRIPT) files. This option is valid only if the user is viewing a multipage document.

Edit

The Edit menu contains the following items.

Undo

If the user has selected one of the various operations on the palette, he may undo his last selection by choosing this menu item.

Palette...

Brings up a palette with various operations that may be performed on the currently displayed image, such as rotation, zooming and flipping.

Properties...

Brings up a dialog that allows the user to specify options that affect how **imagetool** is run. These include whether to display images in gray scale or color, number of colors to display (this option depends on the framebuffer of the users system) and whether or not the palette is to display immediately or not.

Help...

Launches the **helpviewer(1)** which provides the user with **imagetool** online help.

There are two other buttons in the main panel with forward and backward arrows on them. These buttons become active when a multipage document is loaded. Using these buttons, the user may page forward or backward.

At the far right of the main panel, there is a drop target which has two functions. First, the user may drag files from another application (**filemgr(1)** for example) and drop them in the drop target. This causes them to be loaded into the **imagetool** and displayed. Alternately, after a file has been loaded into the **imagetool** for display, the user may drag a copy of the file out of **imagetool** by moving the mouse pointer to the drop target, pressing down on the left mouse button and dragging the resulting file image to another application (**filemgr** again for example).

Below the main panel, there is a canvas on which the image is displayed. When **imagetool** starts up, it tries to create the window to fit the size of the image. However, if the image is very large, it may only show a portion of it. The user may however, use the scrollbars attached to the display canvas to move around within the image.

OPTIONS

- v** Prints out the current version of **imagetool**.
- verbose** Prints lots of debugging information (not useful to the user)
- timeout** *seconds*
Set the timeout value for the Display POSTSCRIPT server. The default value is 60 seconds.
- usage** Prints out valid command line options.

If **imagefile** is specified, image is displayed automatically when the window appears. If no argument is given, **imagetool** comes up with no document or image in it.

RESOURCES

On startup, imagetool will use the following X resources which are stored in \$HOME/.desksetdefaults. Note these resource names will be prepended with deskset.imagetool.

Resource: ViewImageIn
Values: Color, GrayScale (Color)
Description: Determines if images will be displayed in color or grayscale. If user is using a monochrome monitor, this setting has no affect.

Resource: Colors
Values: BW, 16, 256, Millions (256)
Description: Sets the number of colors to be used when viewing images. This resource only is used if a multi-plane framebuffer is being used. Note that for most color monitors, only 256 colors are possible.

Resource: DisplayPalette
Values: True, False (True)
Description: Determines if palette is automatically displayed when first image is opened by imagetool.

Resource: UseDSC
Values: True, False (False)
Description: Determines how POSTSCRIPT documents are interpreted. A well written POSTSCRIPT file contains Document Structuring Comments which separate various sections of the document. If the comments are used correctly, then it is easy to determine where each page of the document begins and ends. By default, **imagetool** does not look for these comments when determining pages. Because of this, backward paging can be slow since the only way to verify that the displayed page will look correct is to begin at the beginning of the document and render pages

until the desired page is found. If the user finds that the performance is not acceptable, then this resource may be set, at which time, **imagetool** will use the Document Structuring Comments to determine where each page begins and ends. Note that this may help performance, but that displayed pages may not look correct due to no comments being found in the document, or the comments being used incorrectly.

SEE ALSO **helpviewer(1)**, **filemgr(1)**, **dps(7)**
Solaris User's Guide

TRADEMARK POSTSCRIPT is a registered trademark of Adobe Systems Inc

| | |
|---------------------|--|
| NAME | imake – C preprocessor interface to the make utility |
| SYNOPSIS | imake [-Ddefine] [-Idir] [-Ttemplate] [-f filename] [-s filename] [-e] [-v] |
| DESCRIPTION | imake is used to generate Makefiles from a template, a set of cpp(1) macro functions, and a per-directory input file called an Imakefile . This allows machine dependencies (such as compiler options, alternate command names, and special make(1S) rules) to be kept separate from the descriptions of the various items to be built. |
| OPTIONS | The following command line options may be passed to imake : <ul style="list-style-type: none"> -Ddefine This option is passed directly to cpp. It is typically used to set directory-specific variables. For example, the X Window System uses this flag to set <i>TOPDIR</i> to the name of the directory containing the top of the core distribution and <i>CURDIR</i> to the name of the current directory, relative to the top. -Idirectory This option is passed directly to cpp(1). It is typically used to indicate the directory in which the imake template and configuration files may be found. -Ttemplate This option specifies the name of the master template file (which is usually located in the directory specified with <i>-I</i>) used by cpp. The default is Imake.tmpl. -f filename This option specifies the name of the per-directory input file. The default is Imakefile. -s filename This option specifies the name of the make description file to be generated but make should not be invoked. If the <i>filename</i> is a dash (-), the output is written to stdout. The default is to generate, but not execute, a Makefile. -e This option indicates the imake should execute the generated Makefile. The default is to leave this to the user. -v This option indicates that imake should print the cpp command line that it is using to generate the Makefile. |
| HOW IT WORKS | imake invokes cpp with any -I or -D flags passed on the command line and passes it the following 3 lines: <pre style="margin-left: 40px;">#define IMAKE_TEMPLATE "Imake.tmpl" #define INCLUDE_IMAKEFILE "Imakefile" #include IMAKE_TEMPLATE</pre> <p>where <i>Imake.tmpl</i> and <i>Imakefile</i> may be overridden by the -T and -f command options, respectively. If the Imakefile contains any lines beginning with a '#' character that is not</p> |

followed by a **cpp** directive (**#include**, **#define**, **#undef**, **#ifdef**, **#else**, **#endif**, or **#if**), **imake** will make a temporary **makefile** in which the '#' lines are prepended with the string `"/**/"` (so that **cpp** will copy the line into the **Makefile** as a comment).

The **Imakefile** reads in file containing machine-dependent parameters (specified as **cpp** symbols), a site-specific parameters file, a file containing **cpp** macro functions for generating **make** rules, and finally the **Imakefile** (specified by `INCLUDE_IMAKEFILE`) in the current directory. The **Imakefile** uses the macro functions to indicate what targets should be built; **imake** takes care of generating the appropriate rules.

The rules file (usually named **Imake.rules** in the configuration directory) contains a variety of **cpp** macro functions that are configured according to the current platform. **imake** replaces any occurrences of the string `"@@"` with a newline to allow macros that generate more than one line of **make** rules. For example, the macro

```
#define    program_target(program, objlist)           @@\
program:  objlist                                   @@\
          $(CC) -o $@ objlist $(LDFLAGS)
```

when called with `program_target(foo, foo1.o foo2.o)` will expand to

```
foo:      foo1.o foo2.o
          $(CC) -o $@ foo1.o foo2.o $(LDFLAGS)
```

On systems whose **cpp** reduces multiple tabs and spaces to a single space, **imake** attempts to put back any necessary tabs (**make** is very picky about the difference between tabs and spaces). For this reason, colons (:) in command lines must be preceded by a backslash (\).

USE WITH THE X WINDOW SYSTEM

The X Window System uses **imake** extensively, for both full builds within the source tree and external software. As mentioned above, two special variables, `TOPDIR` and `CURDIR` set to make referencing files using relative path names easier. For example, the following command is generated automatically to build the **Makefile** in the directory **lib/X/** (relative to the top of the sources):

```
% ../../../../config/imake -I../../../../config \
-DTOPDIR=../../../../ -DCURDIR=./lib/X
```

When building X programs outside the source tree, a special symbol **UseInstalled** is defined and `TOPDIR` and `CURDIR` are omitted. If the configuration files have been properly installed, the script **xmkmf(1)** may be used to specify the proper options:

```
% xmkmf
```

The command **make Makefiles** can then be used to generate **Makefiles** in any subdirectories.

| | | |
|------------------------------|---|---|
| FILES | usr/tmp/tmp-imake.nnnnnn usr/tmp/tmp-make.nnnnnn usr/ccs/lib/cpp | temporary input file for cpp temporary input file for make default C preprocessor |
| SEE ALSO | make(1S) , execvp(2) , xmkmf(1) S. I. Feldman <i>Make – A Program for Maintaining Computer Programs</i> | |
| ENVIRONMENT VARIABLES | The following environment variables may be set, however their use is not recommended as they introduce dependencies that are not readily apparent when imake is run: | |
| | IMAKEINCLUDE If defined, this should be a valid include argument for the C preprocessor. E.g. <i>-I /usr/include/local</i> . Actually, any valid cpp argument will work here. | |
| | IMAKECPP If defined, this should be a valid path to a preprocessor program. E.g. <i>/usr/local/cpp</i> . By default, imake will use <i>/lib/cpp</i> . | |
| | IMAKEMAKE If defined, this should be a valid path to a make program. E.g. <i>/usr/local/make</i> . By default, imake will use whatever make program is found using execvp(2) . | |
| BUGS | Comments should be preceded by “ <i>/**/#</i> ” to protect them from cpp . | |
| AUTHOR | Todd Brunhoff, Tektronix and MIT Project Athena; Jim Fulton, MIT X Consortium | |

| | |
|--------------------|--|
| NAME | kbd_mode – change the keyboard translation mode |
| SYNOPSIS | kbd_mode -a -n -e -u |
| DESCRIPTION | <p>kbd_mode sets the translation mode of the console's keyboard (/dev/kbd) to one of the four values defined for KIOCTRANS in kb(7). This is useful when a program that resets the translation mode terminates abnormally and fails to restore the original translation mode.</p> <p>Note that SunView desires translated events (kbd_mode -e), while Xsun(1) desires untranslated events (kbd_mode -u). See below for an explanation of the -e and -u options.</p> |
| OPTIONS | <p>-a ASCII: the keyboard will generate simple ASCII characters.</p> <p>-n None: the keyboard will generate unencoded bytes – a distinct value for up and down on each switch on the keyboard.</p> <p>-e Events: the keyboard will generate SunWindows input events with ASCII characters in the <i>value</i> field.</p> <p>-u Unencoded: the keyboard will generate input events with unencoded bytes in the <i>value</i> field such as those desired by the Xsun(1) server.</p> |
| FILES | <p>/dev/kbd \$OPENWINHOME/bin/kbd_mode</p> |
| SEE ALSO | kb(7) |

| | |
|--------------------|---|
| NAME | keytablemap – maps keyboard type and layout to keytable |
| SYNOPSIS | \$OPENWINHOME/etc/keytables/keytable.map |
| DESCRIPTION | <p>The keytable map contains information that maps keyboard type and layout to the appropriate keytable file. The keyboard type and layout are generally obtained from the kernel using appropriate ioctl calls. The keyboard layout is assumed to be US layout (code=0) on Sun keyboards prior to type 4. The layout is determined by a keyboard DIP switch setting on type-4 and subsequent types. The keytable file contains a mapping of keycap symbols to physical keys.</p> <p>For each keyboard type and layout, a single line should be present with the following information:</p> <pre style="margin-left: 40px;">keyboardtype keyboardlayout keytablefilename</pre> <p>Any characters following a white space after the keytablefilename field, through the end of line are disregarded. The fields are separated by any number of blanks and/or TAB characters. A leading '#' indicates the beginning of a comment, which may appear only as the first character on the line. Such comment lines are ignored by routines that read the file.</p> |
| EXAMPLE | <p>An example of a keytablemap file appears below.</p> <pre style="margin-left: 40px;">{ # keytable.map # # The keytable map associates a keyboard with its # corresponding keytable. Keytables are located in # \$OPENWINHOME/etc/keytables. # # The keyboard type and layout are generally obtained # from the kernel using appropriate ioctl calls. The # keyboard type is stored in a keyboard ROM. The # keyboard layout is assumed to be 0 on Sun keyboards # prior to type 4 and is determined by a DIP switch # setting on type-4 and later keyboards. # # Notes: # To test a new keytable before installing it, copy it # to \$HOME/.keytable, start the server and test. # (Remember to remove \$HOME/.keytable after testing.) # # Format of an entry: # keyboard_type keyboard_layout keytable_filename #</pre> |

```
#Type Layout Filename
2 0 US2.kt
3 0 US3.kt
4 0 US4.kt # Sun US Type 4 keyboard
4 2 FranceBelg4.kt
4 3 Canada4.kt
# -- Truncated --
```

FILES **SOPENWINHOME/etc/keytables/keytable.map**

NOTES This file is intended to be used only by the OpenWindows server.

| | |
|--------------------|---|
| NAME | listres – list resources in widgets |
| SYNOPSIS | listres [-all] [-nosuper] [-variable] [-top name] [-format printf-string] |
| DESCRIPTION | The listres program 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. If no specific widgets or the -all switch are given, a two-column list of widget names and their class hierarchies is printed. |
| OPTIONS | listres accepts all of the standard toolkit command line options along with those listed below: -all This option indicates that listres should print information for all known widgets and objects. -nosuper This option indicates that resources that are inherited from a superclass should not be listed. This is useful for determining which resources are new to a subclass. -variable This option indicates that widgets should be identified by the names of the class record variables rather than the class name given in the variable. This is useful for distinguishing subclasses that have the same class name as their superclasses. -top name This option specifies the name of the widget to be treated as the top of the hierarchy. Case is not significant, and the name may match either the class variable name or the class name. The default is "core". -format printf-string This option specifies the <i>printf</i> -style format string to be used to print out the name, instance, class, and type of each resource. |
| SEE ALSO | X11(7) , xrdb(1) , appropriate widget documents |
| BUGS | On operating systems that do not support dynamic linking of run-time routines, this program must have all of its known widgets compiled in. The sources provide several tools for automating this process for various widget sets. |
| COPYRIGHT | Copyright 1989, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium |

| | |
|--------------------|--|
| NAME | locale_env - program for openwin-sys |
| SYNOPSIS | locale_env |
| DESCRIPTION | locale_env provides functionality for OpenWindows system initialization script openwin-sys that is better done via a separate program. locale_env is used by openwin-sys and is not intended to be used directly. |
| SEE ALSO | openwin(1) |

| | |
|--------------------|--|
| NAME | mailp, digestp, filep, newsp, filofaxp, franklinp, timemanp, timesysp – frontends to the mp PostScript pretty printer |
| SYNOPSIS | <p>mailp [<i>options</i>] <i>filename</i> ...</p> <p>newsp [<i>options</i>] <i>filename</i> ...</p> <p>digestp [<i>options</i>] <i>filename</i> ...</p> <p>filep [<i>options</i>] <i>filename</i> ...</p> <p>filofaxp [<i>options</i>] <i>filename</i> ...</p> <p>franklinp [<i>options</i>] <i>filename</i> ...</p> <p>timemanp [<i>options</i>] <i>filename</i> ...</p> <p>timesysp [<i>options</i>] <i>filename</i> ...</p> |
| DESCRIPTION | <p>mailp is a frontend to the mp(1) Postscript pretty printer program. It uses different names to provide various mp options:</p> <p>mailp will print out mail messages.</p> <p>newsp will print out USENET news articles.</p> <p>digestp will print out USENET digest files.</p> <p>filep will print out ordinary ASCII files.</p> <p>filofaxp will print out in Filofax personal organiser format.</p> <p>franklinp will print out in Franklin Planner personal organiser format.</p> <p>timemanp will print out in Time Manager personal organiser format.</p> <p>timesysp will print out in Time/System International personal organiser format.</p> <p>mailp (and the associated programs) read each <i>filename</i> in sequence and generate a prettified version of the contents in PostScript format. If no filename arguments are provided, mailp reads the standard input.</p> <p>mailp uses the PRINTER environment variable to determine which printer the output from the mp(1) program is sent to. If this environment variable is not found, then it defaults to the <i>ps</i> printer.</p> |
| OPTIONS | <p>-F Instead of printing who the mail article is for, the top header will contain who the mail article is from. A useful option for people with their own personal printer.</p> <p>-P printer Send output to the named printer. Otherwise send output to the printer named in the PRINTER environment variable,</p> <p>-h Banner printing is disabled. Most of the information that typically appears on the banner sheet is output in the mp banners.</p> <p>-d printer Send output to the named printer. Otherwise send output to the printer named in the PRINTER environment variable,</p> |

-l Format output in landscape mode. Two pages of text will be printed per sheet of paper.

-s *subject*

Use *subject* as the new subject for the printout. If you are printing ordinary ASCII files which have been specified on the command line, the the subject will default to the name of each of these files.

SEE ALSO

mp(1)

AUTHORS

Original version by Bruno Pillard - October 1988.

Modified by Rich Burrige - June 1989 and October 1990.

Revisited by Larry W. Virden - August, September 1990.

Revisited by Bruno Pillard - September 1990.

Revisited by David W. Sanderson - August 1991.

| | |
|--------------------|---|
| NAME | mailprint – filter to strip out mail message attachments |
| SYNOPSIS | mailprint [-b] [<i>filename...</i>] |
| DESCRIPTION | <p>The mailprint program reads a <i>filename</i> (which contains one or more mail messages from mail(1) or mailtool(1), and sends to standard output the same messages, minus any attachments that were included in the mail messages. If no <i>filename</i> argument is provided, mailprint reads from the standard input.</p> <p>mailprint will add the following line to the end of each mail message, if any attachments were stripped out:</p> <p style="padding-left: 40px;">This message contains <i>n</i> attachments (not printed)</p> <p>where <i>n</i> is the number of attachments stripped.</p> <p>This program will output nothing if the filename or standard input is not a message. This program's functionality is already provided in mailtool's print option, as well as printtool when you drag a message from mailtool and drop it onto printtool(1).</p> <p>mailprint can be used together with other programs to print out a message. For example:</p> <p style="padding-left: 40px;">example% cat <i>filename</i> mailprint mp lp</p> |
| OPTIONS | -b At the end of each message, a formfeed/pagebreak is added. This is useful for sending multiple mail messages to a printer. |
| SEE ALSO | mail (1), mailtool (1), printtool (1) |

| | |
|---------------------|---|
| NAME | mailtool – OpenWindows interface for the mail program |
| SYNOPSIS | mailtool [-Mx] [-Mi interval] [-Mf mailfile] [<i>generic-tool-arguments</i>] [-v] |
| AVAILABILITY | This command is available with the OpenWindows environment. mailtool uses the OPEN LOOK Graphical User Interface. |
| DESCRIPTION | <p>mailtool is an electronic mail application that uses the standard OpenWindows interface. It provides a menu-driven facility for reading, storing, composing, and sending mail messages. Scrollable windows allow easy access to the In-Box mail files.</p> <p>The full editing capabilities of textedit(1) are available for modifying and composing mail messages as well as text fields within command panels.</p> |
| OPTIONS | <p>-Mx Expert mode. Do not ask for confirmation after potentially damaging mail commands. This has the same effect as setting the expert variable.</p> <p>-Mi interval Check for new mail every <i>interval</i> seconds. This has the same effect as specifying a value for the interval variable.</p> <p>-Mf mailfile Start up mailtool with <i>mailfile</i> loaded instead of the In-Box.</p> <p>-v Print out the version number of mailtool and exit.</p> <p><i>generic-tool-arguments</i> Mailtool accepts the generic tool arguments described in xview(7).</p> |
| USAGE | <p>mailtool operates through the use of a set of command panel buttons, message windows, menus, and other components that conform to the OPEN LOOK Graphical User Interface Functional Specification. Mail messages are edited using the menus and commands of the textedit program. For more information about the general usage of OPEN LOOK software applications, see the OpenWindows documentation. For more information about textedit conventions, see the textedit man page.</p> <p>Command Panel Buttons</p> <p>The mailtool command panel is located near the top of the main window, under the window header. Press MENU on a command panel button to reveal the menu for that button. Certain menu functions are accelerated, and may be activated directly from the keyboard by holding down the Meta key and the appropriate accelerator key. The accelerated functions are so indicated.</p> <p>File ></p> <p style="padding-left: 40px;">Load In-Box [Meta o] Used to read mail from the user's system mail file into mailtool.</p> <p style="padding-left: 40px;">Print [Meta p] Sends copies of all the selected mail items to your default printer. If there are no selected items, mailtool sends copies of those items</p> |

you are currently viewing.

Save Changes [Meta s]

Causes all the deletions and changes you have made to the mail file to become permanent and opens your **In-Box** for any new mail.

Done

Save your changes and close **mailtool** to an icon. When **mailtool** is next opened, the **In-Box** will be reread.

Mail Files...

Bring up the Mail Files... scrolling list of your mail files. This popup allows you to create, delete, rename, add messages to, and view mail files.

View >

Messages >

Displays messages you have selected.

Abbreviated Header

Strips unnecessary header fields from the messages you view. Specify these fields by adding them to the **Hide** list in the **Message Window** Property Sheet in **mailtool**.

Full Header

Displays the currently selected messages with all of the message header fields.

Previous Determines the message preceding the last one displayed and displays it. The message is chosen from all the messages in the current mail file.

Next Determines the message following the last one displayed and displays it. The message is chosen from all the messages in the current mail file.

Sort By >

Time and Date

Sorts the messages in chronological order.

Sender Sorts the messages alphabetically by sender.

Subject Sorts the messages alphabetically by subject.

Size Sorts the messages by size (smallest to largest).

Status Sorts the messages by status: first read, then unread, and finally new.

Message Number

Sorts the messages by message number, in increasing order.

Find... [Meta f]

From: Enter text that is in the **From** field (mail address) of the message you want to find. Capitalization is ignored.

To/Cc: Enter text that is in the To or Cc field (mail address) of the message you want to find. Capitalization is ignored.

To: Enter text that is in the To field (mail address) of the message you want to find. Capitalization is ignored.

Cc: Enter text that is in the Cc field (mail address) of the message you want to find. Capitalization is ignored.

Subject: Enter text that is in the Subject field of the message you want. Capitalization is ignored.

Find Forward

After entering text for **From:** , **To/Cc:** , **To:** , **Cc:** , **Subject:** , or any combination of the above, click SELECT on **Find Forward** to locate the next message that matches the text.

Find Backward

After entering text for **From:** , **To/Cc:** , **To:** , **Cc:** , **Subject:** or any combination of the above, click SELECT on **Find Backward** to locate the previous message that matches the text.

Select All After entering text for **From:** , **To/Cc:** , **To:** , **Cc:** , **Subject:** or any combination of the above, click SELECT on **Select All** to select all messages that match the text.

Edit >**Cut [Meta x]**

Deletes the selected mail messages, placing copies in the **Clipboard**.

Copy [Meta c]

Copies the selected mail messages, placing copies in the **Clipboard**.

Delete Deletes the selected mail messages, without placing copies in the **Clipboard**. If no mail messages are selected, the messages currently being viewed are deleted.

Undelete >

Last Restores the last message deleted to your mail header display. This may be done until all messages deleted since the last commit are restored. This option also restores messages deleted through cut and move commands.

From List...

Causes a popup window to appear. This popup contains a list box that has all the mail items that have been deleted since the last commit operation. You can select any number of these, and press the **Undelete** button on the command frame. All the selected items will be returned to the mail header window.

Properties... [Meta i]

Bring up the **mailtool** property sheets. You can modify most of the options to **mailtool** through the property sheets.

Compose >

This panel button allows users to create new mail to be sent, reply to existing messages with or without including the current message, and to forward messages as needed.

New [Meta n]

Opens a composition window without the message headers being filled in. If an unused mail message window currently exists on the screen, it will be brought forward to be used and no new window is created. If an unused mail composition window currently exists and is fully displayed on the screen, there will be no change in the display.

Reply >

Opens a composition window with the message headers filled in appropriately and allows you to write and deliver your response. If an unused mail message window currently exists on the screen, it will be brought forward to be used and no new window is created. If an unused mail message window currently exists and is fully displayed on the screen, its headers will be updated. Options to this submenu are described below:

To Sender Opens a composition window. In the window, the address field contains the originator's address for the selected message and the subject field contains the subject line for the selected message, preceded by "Re: ".

To All Opens a composition window. In the window, the address field contains addresses for the person who sent the selected message as well as all the people that the selected message was sent to. The subject field contains the subject line for the selected message, preceded by "Re: ".

To Sender, Include

Opens a composition window. In the window, the address field contains the originator's address for the selected message. The subject field contains the subject line for the selected message, preceded by "Re: ". The

window also contains the selected message in the body of the new message.

To All, Include:

Opens a composition window. In the window, the address field contains addresses for the person who sent the selected message as well as all the people that the selected message was sent to. The subject field contains the subject line for the selected message, preceded by "Re: ". The window also contains the selected message in the body of the new message.

Forward Opens a composition window. In the window, the subject field contains the subject line for the selected message. The window also contains the selected message in the body of the new message, and it contains the attachments (if any) of the selected message in the attachment pane.

Vacation Lets you compose a message that is automatically delivered in response to incoming messages. Use this when you can't read your mail for a period of time and want your message to be responded to automatically. Only one response is sent to each originator over a one-week period.

The Composition Window

This window has its own control panel with the following buttons.

Include >

Bracketed Pastes the currently selected messages into the body of the message you are composing. The included messages are bracketed with special lines.

Indented Pastes the currently selected messages into the body of the message you are composing. Each line of the included message is indented, using the standard indentation string. The indentation string may be modified in the Mail Tool Property Sheet for the Compose Window.

Templates >

Lists the available templates that you can include if any are installed. You can add and remove templates in the Mail Tool Property Sheet for Templates.

Deliver >

Quit window

Mails the message you have composed and then dismisses the compose window.

Close window

Mails the message you have composed, clears the compose

window, and then closes the compose window to an icon.

Clear window

Mails the message and then clears the compose window to prepare it for re-use.

Leave message intact

Mails the message and leaves the message in the compose window.

Headers > Selecting an option affects the headers of the mail message.

Aliases... Bring up the Mail Tool Property Sheet for Aliases. From this property sheet, you can add, delete, or change your local mail aliases.

Add/Delete Bcc:

Adds or deletes the Bcc: line from the message you are composing.

Add/Delete Custom:

Adds or deletes a custom header line from the message you are composing. You will see one menu item for each custom header you have installed. You can install or remove custom headers in the Mail Tool Property Sheet for the Compose Window.

Clear Clears the contents of the compose window.

Attach > Selecting an option allows the creation of a Voice or Appointment attachment.

Voice... Brings up AudioTool for adding audio attachments.

Appt... Brings up Appointment Editor for adding calendar appointment attachments.

Mailtool Variables

In addition to the variables recognized by **mailx(1)**, **mailtool** recognizes those listed below. They can be set by editing your **.mailrc** file; however, since most of the variables are accessible through the Mail Tool Property Sheets, we strongly recommend that you modify them there to reduce the chance of error. Unless otherwise noted, the default for the following variables is *off*.

additionalfields

A list of header fields to access via the **Add Custom** field in the **Header** menu. This variable can be accessed through the **Custom Fields**, **Header Field**, and **Default Value**: portions of the Compose Window category in the Mail Tool Property Sheet.

bell The number of times to ring the bell when new mail arrives. This variable can be accessed through the Mail Tool Property Sheet in the Header Window category as **Signal With**: __ Beep(s). The default is 0.

dontlogmessages

This variable controls whether or not the **log** checkbox is checked in the Compose Message window. It is ignored if the *record* variable is not set. The default is to log messages.

- expert** Set expert mode in which minimal confirmations are requested. This variable can be accessed through the **Request confirmations** check box in the Compose Window Category of the Mail Tool Property Sheet.
- filemenu** A list of files from which to initialize the **Move, Copy, and Load** menus. These can be absolute pathnames or pathnames relative to the directory specified in the *folder* variable. This variable is superceded by the **filemenu2** variable.
- filemenu2** Same as filemenu, but if both exist, filemenu2 takes precedence. This variable can be accessed through the **Move, Copy, Load Menus:** scrolling list and the **Permanent File:** text field in the Mail Filing category of the Mail Tool Property Sheet.
- filemenu2size** Specifies the maximum number of entries in the **Move, Copy, and Load** menus. This variable can be accessed through the **Display Up To:** __ Files in Menus entry in the Mail Filing Category of the Mail Tool Property Sheet. The default is 10.
- flash** The number of times to flash the window or icon when new mail arrives. This variable can be accessed through the Mail Tool Property Sheet in the Header Window Category as **Signal With:** __ Flash(es). The default is 0.
- folder** The directory for saving mail files. This variable can be accessed as **Mail File Directory** in the Mail Filing category of the Mail Tool Property Sheet.
- headerlines** The number of lines to display at a time in the header window. This variable can be accessed through the Mail Tool Property Sheet in the Header Window category as **Display** __ Headers. The default is 15.
- hideattachments** Hide the attachments pane in the Compose Message window. This variable can be accessed through the **Defaults:** Show attachment list check box in the Compose Window category of the Mail Tool Property Sheet. The default is to show the attachment pane.
- indentprefix** When indentprefix is set, the string that it is set to is used to mark indented lines from included messages. The default indentprefix is ">".
- popuplines** The number of lines in the View Message and Compose Message Windows. This variable can be accessed through the Mail Tool Property Sheet in the Message Window category as **Display** __ Lines of Text. The default is 30.
- printmail** The command to use to print a message. This variable can be accessed through the Mail Tool Property Sheet in the Message Window category as **Print Script**. The default is **lp -s**.
- record** The mail file in which to record outgoing messages. If *record* is set, a **Log** check box will appear on the Compose Message window. If the check box is

checked, the message will be logged in the record file when it is sent. If it is not checked, the message will not be logged. The *dontlogmessages* variable controls whether or not the check box is checked by default. The *record* variable may be set through the **Logged Messages File** item in the Compose Window category of the Mail Tool Property Sheet.

retrieveinterval

The interval in seconds to check for new mail. This variable can be accessed through the Mail Tool Property Sheet in the Header Window category as **Retrieve Every *n* Seconds**. The default is 300.

save Save contents of each Compose Message window in a dead.letter file until the message is delivered successfully. If a Compose Message window is quit, and a new one is brought up, the new window will reuse the dead.letter from the previous window. The first dead.letter file is called dead.letter, the second one is called dead.letter.1, the third dead.letter.2, and so on. The default is *on*.

showto Show the "To" field of mail messages in the Header Window if the mail is from the same user that is reading mail (eg. you).

sortfilemenu

Sort the **Move**, **Copy**, and **Load** menus alphabetically.

suppressautoretrieve

Do not automatically retrieve new mail messages. This variable can be accessed through the Mail Tool Property Sheet in the Header Window category as the Automatically display headers check box. Default is to automatically retrieve new mail.

templates A list of *name:path* pairs to access via the **Include > Templates** menu. *name* appears in the menu; *path* is the file included when name is selected. This variable can be accessed in the Template category of the Mail Tool Property Sheet. By default, the calendar template is installed.

toolcols Default width of Mail Tool windows (in columns). This variable can be accessed through the Mail Tool Property Sheet in the Header Window category as **Display: __ Characters wide**. Default is 80.

trash The name of the trash bin, which may be accessed just like any other mail file. If set, all deleted messages are moved to the trash bin. The trash bin is emptied when you commit changes. This option degrades the performance of **mailtool** and is not recommended.

Mail Tool Commands

In addition to the commands recognized by **mailx(1)** in the *.mailrc* file, **mailtool** also recognizes the following commands.

#-button

This command is used to create the four custom buttons in **mailtool**. It can be accessed through the **Custom Buttons:**, **Command:**, and **Label:** properties in the Header Window category of the Mail Tool Property Sheet.

#-clearaliases

This command is used to clear all the aliases defined above the current line in the .mailrc file. Mail Tool uses this command to help prevent aliases from being defined twice when it saves alias definitions to the .mailrc file.

ignore [header-field...]

Suppress displaying of the specified header fields. Examples of header fields to ignore are Status and Received. The fields are also ignored when the message is saved or printed. This variable can be accessed through the Mail Tool Property Sheet in the Message Window category by using the **Hide:** scrolling list and the **Header Field:** text field.

The .mailtool-init File

The **.mailtool-init** file is created in your home directory when a "Save Workspace" command is invoked from the Workspace menu. This file contains the current positions and sizes of mailtool's View, Compose, and Header Windows (i.e., what's currently on your screen), so on the next invocation of **mailtool** your mailtool windows will start-up with the same layouts.

The following 3 lines show the syntax of .mailtool-init's contents:

viewwin **xloc** *number* **yloc** *number* **width** *number* **height** *number*

compwin [**iconic**] **xloc** *number* **yloc** *number* **width** *number* **height** *number* **ixloc** *number* **iyloc** *number* **deldef** *number*

basewin **filedef** *number* **viewdef** *number* **editdef** *number* **compdef** *number* **repdef** *number*

The first line specifies the x and y coordinates of the top left corner of the View Window, and its width and height (all in pixels).

The second line specifies the same information for the Compose Window. In addition, the **iconic** word, if present, specify that the Compose Window will come up as an icon in the positions specified by the **ixloc** and **iyloc** parameters. The **deldef** parameter specifies the menu default of the **Deliver** button (the number 1 for the first menu item, etc.).

The third line specifies, for the Header Window, menu default items for the **File**, **View**, **Edit**, **Compose**, and **Reply** menus, respectively.

The .mtdeletelog File

Mailtool creates the **.mtdeletelog** file in your home directory so that it can recover the undelete list if **mailtool** is terminated abnormally while reading the spool file. The next time **mailtool** comes up after an abnormal termination, it will remember which files were deleted so you don't have to delete them again, and the undelete list is there so that you can undelete any of those messages before

saving your changes. This only works if you were editing your In-Box, it doesn't work for other mail files.

Signals

SIGUSR1 If you send this signal to **mailtool**, **mailtool** will act as if you clicked on the **Done** button. It will save any changes, close the mail file, and iconify.

ENVIRONMENT

The following are environment variables taken from the execution environment and are not alterable within **mailtool**.

HOME=*directory*

The user's home directory.

MAIL=*filename*

The name of the initial mailbox file to read (in lieu of the standard system mailbox). The default is **/var/mail/username**.

MAILRC=*filename*

The name of the start-up file. Default is **\$HOME/.mailrc**.

FILES

/var/mail/* System mailboxes

/etc/mail/mailx.rc

System setup file that is read in before **~/mailrc**.

~/mailrc

Start-up file for **mail** and **mailtool**.

~/mailtool-init

Start-up file for **mailtool** that contains the positions and sizes of **mailtool**'s View, Compose, and Header Windows.

~/mtdeletelog

File **mailtool** uses to keep track of deleted messages.

SEE ALSO

mail(1), **mailx(1)**, **newaliases(1)**, **sendmail(1M)**, **textedit(1)**, **vacation(1)**, **aliases(4B)**, **xview(7)**

OpenWindows user documentation

"About Mail Tool" in the Help Handbook available through the Help option on the Workspace menu.

| | |
|--------------------|--|
| NAME | makebdf – create bitmap files from scalable F3 or X11/NeWS font files |
| SYNOPSIS | makebdf [-a] [-A] [-e filename] [-f n] [-m -M] [-p -P] [-s] [-v -V] [-values] [-x] |
| DESCRIPTION | <p>makebdf takes a list of Scalable F3 Font Format outlines (.f3b files) or X11/NeWS Bitmap fonts (.fb files) and generates ASCII bitmap fonts in either Bitmap Distribution Format (.bdf) or the Adobe Font Bitmap (.afb) format and Adobe Font Metrics (.afm) files.</p> <p>makebdf will not create bitmaps if screen protection for the F3 font has been activated or if any form of protection exists and the requested size is 24 or higher.</p> |
| NOTES | Options with * apply to F3 files only, they are ignored for *.fb files. |
| OPTIONS | <p>-a Produce font metric (.afm) files only.*</p> <p>-A Produce Adobe Font Bitmap (.afb) format files.</p> <p>-e filename Specify an encoding map file to be loaded along with the font. This option is only required if the encoding specified by the font is other than ISO Latin-1, Symbol or Dingbats.</p> <p>-fn Force the length of the base part of the output filename to be at most <i>n</i> characters. The default is 32.</p> <p>-m Enable generation of .afm files (the default).*</p> <p>-M Disable generation of .afm files.*</p> <p>-p Preserve existing files. If -p is selected, then just before makebdf writes a file it will check to see if it already exists. If it does, the file will be skipped. This is useful in situations where you have some handbuilt .afb and .afm files, and just want to fill in the missing ones.</p> <p>-P Don't preserve existing files (the default).</p> <p>-s Produce a synthetic encoding map file in <i><encodingname>.map</i>.*</p> <p>-v Verbose: print messages indicating what's going on (the default).</p> <p>-V Work silently.</p> |

- sizes** A comma separated list of pixel sizes for which **.afb** files should be generated. The default is *6,8,10,12,14,16,18*.*
- x** Produce bitmap fonts in BDF2.1 format. Bitmap font file suffix will be **.bdf**. This is the default.

EXAMPLES

```
example% makebdf *.f3b
```

```
example% makebdf -e latin.map *.fb
```

```
example% makebdf -p -4,5,6,7,8,9,10,11,12,14,16,18,20,24 *.f3b
```

| | |
|--------------------|---|
| NAME | makedepend – create dependencies in makefiles |
| SYNOPSIS | makedepend [-D <i>name=def</i>] [-D <i>name</i>] [-I <i>includedir</i>] [-f <i>makefile</i>] [-o <i>objsuffix</i>] [-s <i>string</i>] [-w <i>width</i>] [- <i>otheroptions</i>] <i>sourcefile</i> |
| DESCRIPTION | <p>Makedepend reads each <i>sourcefile</i> in sequence and parses it like a C-preprocessor, processing all <i>#include</i>, <i>#define</i>, <i>#undef</i>, <i>#ifdef</i>, <i>#ifndef</i>, <i>#endif</i>, <i>#if</i> and <i>#else</i> directives so that it can correctly tell which <i>#include</i> directives would be used in a compilation. Any <i>#include</i> directives can reference files having other <i>#include</i> directives--parsing will occur in these files as well.</p> <p>Every file that a <i>sourcefile</i> includes, directly or indirectly, is what makedepend calls a <i>dependency</i>. These dependencies are then written to a makefile in such a way that make(1S) will know which object files must be recompiled when a dependency has changed.</p> <p>By default, makedepend places its output in the file named makefile if it exists, otherwise Makefile. An alternate makefile may be specified with the -f option. It first searches the makefile for the line</p> <pre style="margin-left: 40px;"># DO NOT DELETE THIS LINE -- make depend depends on it.</pre> <p>or one provided with the -s option, as a delimiter for the dependency output. If it finds it, it will delete everything following this to the end of the makefile and put the output after this line. If it doesn't find it, the program will append the string to the end of the makefile and place the output following that. For each <i>sourcefile</i> appearing on the command line, makedepend puts lines in the makefile of the form</p> <pre style="margin-left: 40px;"><i>sourcefile.o</i>: <i>dfile</i> ...</pre> <p>Where <i>sourcefile.o</i> is the name from the command line with its suffix replaced with .o, and <i>dfile</i> is a dependency discovered in a <i>#include</i> directive while parsing <i>sourcefile</i> or one of the files it included.</p> |
| EXAMPLE | <p>Normally, makedepend will be used in a makefile target so that typing "make depend" will bring the dependencies up to date for the makefile. For example,</p> <pre style="margin-left: 40px;">SRCS = file1.c file2.c ... CFLAGS = -O -DHACK -I../foobar -xyz depend: makedepend -- \$(CFLAGS) -- \$(SRCS)</pre> |
| OPTIONS | <p>Makedepend will ignore any option that it does not understand so that you may use the same arguments that you would for cc(1B).</p> <p>-D <i>name=def</i> or -D <i>name</i> Define. This places a definition for <i>name</i> in makedepend's symbol table. Without <i>=def</i> the symbol becomes defined as "1".</p> <p>-I <i>includedir</i> Include directory. This option tells makedepend to prepend <i>includedir</i> to</p> |

- its list of directories to search when it encounters a *#include* directive. By default, **makedepend** only searches **/usr/include**
- f** *makefile* Filename. This allows you to specify an alternate **makefile** in which **makedepend** can place its output.
 - o** *objsuff* Object file suffix. Some systems may have object files whose suffix is something other than ".o". This option allows you to specify another suffix, such as ".b" with *-o.b* or ".obj" with *-o.obj* and so forth.
 - s** *string* Starting string delimiter. This option permits you to specify a different string for **makedepend** to look for in the makefile.
 - w** *width* Line width. Normally, **makedepend** will ensure that every output line that it writes will be no wider than 78 characters for the sake of readability. This option enables you to change this width.
 - – *options* – – If **makedepend** encounters a double hyphen (– –) in the argument list, then any unrecognized argument following it will be silently ignored; a second double hyphen terminates this special treatment. In this way, **makedepend** can be made to safely ignore esoteric compiler arguments that might normally be found in a CFLAGS **make** macro (see the **EXAMPLE** section above). All options that **makedepend** recognizes and appear between the pair of double hyphens are processed normally.

ALGORITHM

The approach used in this program enables it to run an order of magnitude faster than any other *dependency generator* that all files compiled by a single **makefile** will be compiled with roughly the same **-I** and **-D** options; and that most files in a single directory will include largely the same files.

Given these assumptions, **makedepend** expects to be called once for each **makefile**, with all source files that are maintained by the **makefile** appearing on the command line. It parses each source and include file exactly once, maintaining an internal symbol table for each. Thus, the first file on the command line will take an amount of time proportional to the amount of time that a normal C preprocessor takes. But on subsequent files, if it encounters an include file that it has already parsed, it does not parse it again.

For example, imagine you are compiling two files, **file1.c** and **file2.c**, they each include the header file **header.h**, and the file **header.h** in turn includes the files **def1.h** and **def2.h**. When you run the command

```
makedepend file1.c file2.c
```

makedepend will parse **file1.c** and consequently, **header.h** and then **def1.h** and **def2.h**. It then decides that the dependencies for this file are

```
file1.o: header.h def1.h def2.h
```

But when the program parses **file2.c** and discovers that it, too, includes **header.h**, it does not parse the file, but simply adds **header.h**, **def1.h** and **def2.h** to the list of dependencies

for **file2.o**.

SEE ALSO **cc(1B)**, **make(1S)**

BUGS If you do not have the source for **cpp**, the Berkeley C preprocessor, then **makedepend** will be compiled in such a way that all **#if** directives will evaluate to "true" regardless of their actual value. This may cause the wrong **#include** directives to be evaluated. **Mak-edepend** should simply have its own parser written for **#if** expressions.

Imagine you are parsing two files, **file1.c** and **file2.c**, each includes the file **def.h**. The list of files that **def.h** includes might truly be different when **def.h** is included by **file1.c** than when it is included by **file2.c**. But once **makedepend** arrives at a list of dependencies for a file, it is cast in concrete.

AUTHOR Todd Brunhoff, Tektronix, Inc. and MIT Project Athena

| | |
|--------------------|--|
| NAME | makepsres – build PostScript resource database file. |
| SYNOPSIS | makepsres [<i>options</i>] <i>directory</i> ... |
| DESCRIPTION | <p>makepsres creates PostScript language resource database files. Resource database files can be used to specify the location of resources that are used by the font selection panel and other Adobe software. For a complete description of the resource location facilities in the Display PostScript system, see Appendix A and Appendix B of "Display PostScript Toolkit for X" in <i>Programming the Display PostScript System with X</i>.</p> <p>makepsres creates a resource database file named PSres.upr that contains all the resources in all the <i>directory</i> path names specified on the command line.</p> <p>If the list of directories contains -, makepsres reads from <i>stdin</i> and expects a list of directories separated by space, tab, or newline.</p> <p>If the list of directories is empty, it is taken to be the current directory.</p> <p>If all specified directories have a common initial prefix, makepsres extracts it as a directory prefix in the new resource database file.</p> <p>makepsres normally acts recursively; it looks for resource files in subdirectories of any specified directory. This behavior can be overridden with the command line option -nr.</p> <p>makepsres uses existing resource database files to assist in identifying files. By default, makepsres creates a new resource database file containing all of the following that apply:</p> <ul style="list-style-type: none">Resource files found in the directories on the command line.Resource files pointed to by the resource database files in the directories on the command line.Resource entries found in the input resource database files. These entries are copied if the files they specify still exist and are located in directories not specified on the command line. <p>If you run makepsres in discard mode (with the -d option), it does not copy resource entries from the input resource database files. In that case, the output file consists only of entries from the directories on the command line. The input resource database files are only used to assist in identifying files.</p> <p>If you run makepsres in keep mode (with the -k option), it includes in the output file all resource entries in the input resource database files, even entries for files that no longer exist or are located in directories specified on the command line.</p> <p>makepsres uses various heuristics to identify files. A file that is of a private resource type or that does not conform to the standard format for a resource file must be specified in one of the following ways:</p> <ul style="list-style-type: none">By running makepsres in interactive modeBy preloading the file into a resource database file used for input |

By beginning the file with the following line:

```
%!PS-Adobe-3.0 Resource-<resource-type>
```

OPTIONS

-o filename

Writes the output to the specified filename. The construction "**-o -**" writes to stdout. If the **-o** option is not specified, **makepsres** creates a **PSres.upr** file in the current directory and writes the output to that file.

-f filename

Uses information from the specified file to assist in resource typing. The file must be in resource database file format. Multiple **-f** options may be specified. The construction "**-f -**" uses *stdin* as an input file and may not be used if "**-**" is specified as a directory on the command line.

-dir dirname

Specifies that *dirname* is a directory. Needed only in rare cases when *dirname* is the same as a command-line option such as **-nb**.

-d

Specifies discard mode. The resulting output file consists solely of entries from the directories on the command line.

-e

Marks the resulting **PSres.upr** file as exclusive. This option makes the resource location library run more quickly since it does not have to look for other resource database files. It becomes necessary, however, to run **makepsres** whenever new resources are added to the directory, even if the resources come with their own resource database file.

-i

Specifies interactive mode. In interactive mode, you will be queried for the resource type of any encountered file that **makepsres** cannot identify. If **-i** is not specified, **makepsres** assumes an unidentifiable file is not a resource file.

-k

Specifies keep mode.

-nb

If the output file already exists, do not back it up.

-nr

Specifies nonrecursive mode. **makepsres** normally acts recursively: it looks for resource files in subdirectories of any specified directory. If **-nr** is used, **makepsres** does not look in subdirectories for resource files.

-p

Specifies no directory prefix. If **-p** is used, **makepsres** does not try to find a common directory prefix among the specified directories.

-q

Quiet mode: ignores unidentifiable files instead of warning about them.

-s

Specifies strict mode. If **-s** is used, **makepsres** terminates with an error if it encounters a file it cannot identify.

EXAMPLES

example% makepsres .

Creates a resource database file that contains all the resources in the current directory.

example% makepsres -i -o local.upr /usr/local/lib/ps/fonts

Runs **makepsres** in interactive mode and creates a resource database file named **local.upr**, which contains all the resources in the directory **/usr/local/lib/ps/fonts**.

SEE ALSO *Programming the Display PostScript System with X* (Addison-Wesley Publishing Company, Inc., 1993).

AUTHOR Adobe Systems Incorporated

NOTES PostScript and Display PostScript are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.
Copyright (c) 1989-1991 Adobe Systems Incorporated. All rights reserved.

| | |
|--------------------|---|
| NAME | maze – an automated maze program |
| SYNTAX | maze [-S] [-r] [-g geometry] [-d display] |
| DESCRIPTION | The maze program creates a "random" maze and then solves it with graphical feedback. |
| OPTIONS | <p>-S Full screen window option...</p> <p>-r Reverse video option...</p> <p>-g geometry Specifies the window geometry to be used...</p> <p>-d display Specifies the display to be used...</p> <p>The following lists the current functionality of various mouse button clicks:</p> <p>LeftButton Clears the window and restarts maze...</p> <p>MiddleButton Toggles the maze program, first click -> <i>stop</i>, second click -> <i>continue</i>...</p> <p>RightButton Kills maze...</p> |
| LIMITATIONS | <p>No color support...</p> <p>Expose events force a restart of maze...</p> <p>Currently, mouse actions are based on "raw" values [Button1, Button2 and Button3] from the ButtonPress event...</p> <p>[doesn't use pointer mapping]</p> |
| COPYRIGHT | <p>Copyright 1988 by Sun Microsystems, Inc. Mountain View, CA.</p> <p>All Rights Reserved</p> <p>Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of Sun or MIT not be used in advertising or publicity pertaining to distribution of the software without specific prior written permission. Sun and M.I.T. make no representations about the suitability of this software for any purpose. It is provided "as is" without any express or implied warranty.</p> <p>SUN DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SUN BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.</p> |

| | | |
|------------------|-------------------------------------|---------------------------|
| AUTHOR(s) | Richard Hess [X11 extensions] | {...}!uunet!cimshop!rhess |
| | Consilium, Mountain View, CA | |
| | Dave Lemke [X11 version] | lemke@sun.COM |
| | Sun Microsystems, Mountain View, CA | |
| | Martin Weiss [SunView version] | |
| | Sun Microsystems, Mountain View, CA | |

| | |
|--------------------|--|
| NAME | mkdirhier – makes a directory hierarchy |
| SYNOPSIS | mkdirhier directory ... |
| DESCRIPTION | The mkdirhier command creates the specified directories. Unlike mkdir if any of the parent directories of the specified directory do not exist, it creates them as well. |
| SEE ALSO | mkdir (1) |

| | |
|--------------------------|--|
| NAME | mkfontdir – create fonts.dir file from directory of font files |
| SYNOPSIS | mkfontdir [<i>directory-names</i>] |
| DESCRIPTION | <p>For each directory argument, mkfontdir reads all of the font files in the directory searching for properties named "FONT", or (failing that) the name of the file stripped of its suffix. These are used as font names, which are written out to the file "fonts.dir" in the directory along with the name of the font file.</p> <p>The kinds of font files read by mkfontdir depends on configuration parameters, but typically include PCF (suffix ".pcf"), SNF (suffix ".snf"), BDF (suffix ".bdf"), F3 (suffix ".f3b") and F3 bitmap (suffix ".fb"). If a font exists in multiple formats, mkfontdir will first choose fonts in fonts in the following order: F3, PCF, SNF, BDF and finally F3 bitmap (.fb).</p> |
| SCALABLE FONTS | <p>Because scalable font files do not usually include the X font name, the fonts.dir file in directories containing such fonts must be edited by hand to include the appropriate entries for those fonts. However, when mkfontdir is run, all of those additions will be lost, so be careful.</p> <p>There is an alternative to editing "fonts.dir" file by hand. If there exists a "fonts.scale" file in any directory of the font-path, mkfontdir will copy the contents of this file to "fonts.dir" file. Any changes can therefore be safely made to "fonts.scale" and when mkfontdir is run in that directory, the changes will be reflected in the "fonts.dir" file also.</p> <p>For example, the directory /usr/openwin/lib/X11/fonts/Speedo contains a "fonts.scale" file, which is hand created. The format of a "fonts.scale" file is as follows:</p> <pre style="margin-left: 40px;"> <num-of-entries> filename XLFD name </pre> <p>Note: F3 scalable fonts do contain enough information for mkfontdir to be able to construct the X font name (XLFD name). Therefore, the above does not apply to F3 format font files.</p> |
| FONT NAME ALIASES | <p>The file "fonts.alias" which can be put in any directory of the font-path is used to map new names to existing fonts, and should be edited by hand. The format is straight forward enough, two white-space separated columns, the first containing aliases and the second containing font-name patterns.</p> <p>When a font alias is used, the name it references is search for in the normal manner, looking through each font directory in turn. This means that the aliases need not mention fonts in the same directory as the alias file.</p> <p>To embed white-space in either name, simply enclose them in double-quote marks, to embed double-quote marks (or any other character), precede them with back-slash:</p> <pre style="margin-left: 40px;"> "magic-alias with spaces" "\"font\name\" with quotes" regular-alias fixed </pre> |

If the string "FILE_NAMES_ALIASES" stands alone on a line, each file-name in the directory (stripped of it's suffix) will be used as an alias for that font.

USAGE Both the **X server** and the **Font Server** look for "fonts.dir" and "fonts.alias" files in each directory in the font path each time it is set (see **xset(1)**).

SEE ALSO **X11(7)**, **Xserver(1)**, **xset(1)**

| | |
|--------------------|--|
| NAME | mp – PostScript pretty printer |
| SYNOPSIS | mp [-A4] [-C] [-F] [-L localename] [-PS] [-US] [-a] [-c chars] [-d] [-e] [-ff] [-fp] [-l] [-m] [-o] [-p prologue] [-s subject] [-tm] [-ts] [-v] [-w words] [-?] [<i>filename...</i>] |
| DESCRIPTION | <p>The mp program reads each <i>filename</i> in sequence and generates a prettified version of the contents in POSTSCRIPT format, sent to standard output. If no filename argument is provided, mp reads the standard input. If the standard input is a terminal, input is terminated by an EOF signal, usually Ctrl-D.</p> <p>Mail items, news articles, ordinary ASCII files, complete mail folders, and digests are all acceptable input formats for mp. The output format includes grayscale lozenges containing banner information at the top and bottom of every page.</p> <p>The program is conveniently used in conjunction with the print button of the mailtool(1) program, or the <i>pipe</i> command provided by mail(1). Add the following two lines to your .mailrc file:</p> <pre style="margin-left: 40px;">set printmail='mp lp' set cmd="mp lp &"</pre> <p>Source the .mailrc file, and you are ready to use mp. For printing ordinary ASCII files, the following alias (to be placed in your</p> <pre style="margin-left: 40px;">alias print 'mp -o -s "!*" <!*" lp'</pre> |
| OPTIONS | <p>-A4 Use A4 paper size (8.5 x 11.4 inches).</p> <p>-C Instead of using "\nFrom" to denote the start of new mail messages, mp will look for (and use) the value of the Content-Length: mail header. If the Content-Length doesn't take you to the next "\nFrom", then it's wrong, and mp falls back to looking for the next "\nFrom" in the mail folder.</p> <p>-F Instead of printing who the mail article is for, the top header will contain who the mail article is from. A useful option for people with their own personal printer.</p> <p>-L localename Provide the locale of the file to be printed. If this command line option is not present, then mp looks for the <i>MP_LANG</i> environment variable. If that is not present, then the <i>LANG</i> environment variable is used. If none of these options are present, then mp tries to determine the locale it is running in, and if it cannot, then it assumes it is running in the C locale, otherwise a prologue file specific to the given locale is prepended to the output. This is to provide I18N level 4 (multi-byte) printing.</p> <p>-PS If the mail or digest message just has PostScript as the text of the message, then this is normally just passed straight through. Specifying this option, causes PostScript to be printed as text.</p> <p>-US Use US paper size (8.5 x 11 inches). This is the default paper size.</p> <p>-a Format the file as a news article. The top banner contains the text: "Article from</p> |

newsgroup", where *newsgroup* is the first news group found on the *Newsgroups:* line.

-c chars

The maximum number of characters to extract from the *gecos* field of the users */etc/passwd* entry. The default is 18.

-d Format the file as a digest.

-e Assume the ELM mail frontend intermediate file format. Used when printing messages from within ELM (using the "p" command), especially for printing tagged messages. This option must be specified in your ELM option setup.

-ff Format the file for use with a Filofax personal organiser.

-fp Format the file for use with a Franklin Planner personal organiser.

-l Format output in landscape mode. Two pages of text will be printed per sheet of paper.

-m Format the file as a mail folder, printing multiple messages.

-o Format the file as an ordinary ASCII file.

-p prologue

Employ the file *prologue* as the POSTSCRIPT prologue file, overriding any previously defined file names.

-s subject

Use *subject* as the new subject for the printout. If you are printing ordinary ASCII files which have been specified on the command line, the the subject will default to the name of each of these files.

-tm Format the file for use with the Time Manager personal organiser.

-ts format the file for use with the Time/System International personal organiser.

-v Print the version number of this release of *mp*.

-w words

The maximum number of words to extract from the *gecos* field of the users */etc/passwd* entry. The default is 3.

-? Print the usage line for **mp** (note that the ? character must be escaped if using **csh(1)**).

**ENVIRONMENT
VARIABLES**

The **mp** prologue file is determined by first looking for the environment variable *MP_PROLOGUE*, which specifies the directory where **mp** prologue files are to be found. If *MP_PROLOGUE* is not found, then the default directory is assumed (*\$OPENWINHOME/share/xnews/client/mp*).

mp also checks for the *MP_LANG* and *LANG* environment variables. If present, then a prologue file called (*\$OPENWINHOME/lib/locale/<localename>/print/prolog.ps*) is prepended to the output to be printed.

**SUPPLIED
PROLOGUE FILES**

The following prologue files are provided:

mp.pro.ps

Used by default

mp.pro.ff.ps

Used if the **-ff** option is in effect

mp.pro.fp.ps

Used if the **-fp** option is in effect

mp.pro.tm.ps

Used if the **-tm** option is in effect

mp.pro.ts.ps

Used if the **-ts** option is in effect

mp.pro.alt.ps

An alternative modification of the default prologue file which outputs the page number in the right corner of the bottom banner.

FILES

.cshrc initialization file for **cs**h(1)

.mailrc initialization file for **mail**(1)

\$OPENWINHOME/bin/mp
executable

\$OPENWINHOME/share/xnews/client/mp/mp.pro.ps
POSTSCRIPT prologue for mail printing

\$OPENWINHOME/share/xnews/client/mp/mp.pro.l.ps
POSTSCRIPT prologue for landscape format

\$OPENWINHOME/share/xnews/client/mp/mp.pro.alt.ps
alternative "default" POSTSCRIPT prologue, inserts page numbers in the bottom right corner of each page

\$OPENWINHOME/share/xnews/client/mp/mp.pro.ff.ps
POSTSCRIPT prologue for Filofax format

LIBDIR/mp.pro.fp.ps
POSTSCRIPT prologue for Franklin Planner format.

\$OPENWINHOME/share/xnews/client/mp/mp.pro.tm.ps
POSTSCRIPT prologue for Time Manager format

\$OPENWINHOME/share/xnews/client/mp/mp.pro.ts.ps
POSTSCRIPT prologue for Time/System International format.

SEE ALSO

mail(1), **mailto**l(1),

AUTHORS

Original version by Steve Holden.

Converted to C, modified and maintained by Rich Burrige, SunSoft Inc, Mountain View.

Original modified to handle net news articles and MH mail by Bruno Pillard, Chorus Systemes, France.

Handling of mail digests added by Dave Glowacki of Public Works Computer Services, St Paul, MN.

Manual page revised by Rick Rodgers, UCSF School of Pharmacy, San Francisco.

Support for Personal Organiser printing style added by Douglas Buchanan, Sun Microsystems Europe.

Substantial modifications to header parsing by Jeremy Webber, Computer Science Department, University of Adelaide, Australia.

Support for printing multiple files and subject line filename print for ordinary ASCII files added by Sam Manoharan, Edinburgh University.

Support for landscape mode written by Michael Tuciarone.

Revision of the POSTSCRIPT structuring and the way that the prologue files are handled was included by Johan Vromans.

New style POSTSCRIPT prologue files by John Macdonald.

Support for the ISO8859 character set by Bertrand DeCouty.

Rich Burrige. MAIL: richb@Eng.Sun.COM

NAME muncher – draw interesting patterns in an X window

SYNOPSIS **muncher** [**-r**] [**-s seed**] [**-v**] [**-q**] [**-geometry geometry**] [**-display display**]

OPTIONS

- r** display in the root window
- s seed** seed the random number seed
- v** run in verbose mode
- q** run in quite mode
- geometry geometry**
define the initial window geometry; see **X11(7)**.
- display display**
specify the display to use; see **X11(7)**.

DESCRIPTION **Muncher** draws some interesting patterns in a window.

SEE ALSO **X11(7)**

COPYRIGHT Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

| | |
|--------------------|---|
| NAME | navigator – browse and search AnswerBook on-line documentation |
| SYNOPSIS | navigator [-b <i>library-file</i>] [-c <i>card-catalog</i>] |
| DESCRIPTION | <p>navigator and docviewer(1) together comprise an OpenWindows application for viewing and navigating AnswerBook on-line document collections.</p> <p>navigator provides three modes for accessing AnswerBook documents: Contents (table-of-contents browsing), Search (full-text search and retrieval), and Bookmarks (links the user creates to frequently referenced pages).</p> <p>Documents found using navigator may then be viewed with docviewer(1).</p> <p>New to this version of navigator is the AnswerBook Library feature. An AnswerBook library is a group of AnswerBooks that can be browsed and searched as a single unit. navigator lets you select AnswerBooks of interest from those available on the network, and add them to the current AnswerBook Library. AnswerBook Library contents are stored in files and thus saved between navigator sessions. In addition, these files may be shared among groups of AnswerBook users, and even mailed to others. See ab_library(4) for more information.</p> <p>You should not start navigator directly. Instead, use the answerbook(1) script, which performs certain operations to initialize and verify the AnswerBook environment before starting navigator.</p> |
| OPTIONS | <p>-b <i>library-file</i> Specify the AnswerBook library to load. The default library is \$HOME/.ab_library. See ab_library(4) for more information.</p> <p>-c <i>card-catalog</i> Specify the name of the card catalog file(s) used to locate AnswerBooks. See ab_cardcatalog(4) for more information.</p> <p>Contents Mode Click SELECT on the "Contents" button near the top of the Navigator window to browse the tables of contents of the AnswerBooks in the current AnswerBook Library.</p> <p>The Location list – the upper scrolling list in Contents mode – shows the current path through the table of contents hierarchy. The Contents (lower) list show the contents of the current title in the location list.</p> <p>Double-click SELECT on any bold-faced title in the Contents list to display that title's contents. The title itself moves to the bottom of the Location list, and the first page of the corresponding document is displayed in the Viewer window. Non-bold-faced titles do not expand, but are displayed in the Viewer.</p> <p>To move back up the table of contents hierarchy, just click SELECT on any title in the Location list. That document's contents will then appear in the Contents list.</p> <p>Search Mode Click SELECT on the "Search" button near the top of the Navigator window to</p> |

perform full-text searches across all the AnswerBooks in the current AnswerBook Library.

The search query syntax is generally free-form (but see below). Simply enter one or more words or phrases in the "Search Library for" text window, then hit RETURN (or click SELECT on "Start Search" button) to initiate the search. The navigator displays the list of titles of documents found, ranked in order of relevance to the search query. The algorithm used to determine document relevance is heuristical, and is based on word-occurrence statistics.

Double-click SELECT on any title in the list to display the corresponding document in the Viewer window.

Search for literal phrases by enclosing them in double quotes:

"workspace properties"

Search for words in proximity by enclosing them in parentheses:

(print mail messages)

Match different work ending with an asterisk:

print* (*matches "print", "printer", "printing", etc.*)

Previous Searches

Click SELECT on "Previous Searches..." to bring up a list of earlier search queries. The Previous Searches window provides a simple copy-and-paste facility for redoing or modifying queries.

Search Settings

Click SELECT on "Search Settings..." to modify search properties. In the Search Settings window, you can select the search scope (either the entire document, or just document titles); the sorting order for the results list (by relevance, or by book and relevance); and the maximum number of titles to display.

Bookmarks Mode

Click SELECT on the "Bookmarks" button near the top of the Navigator window to browse the list of bookmarks in the current AnswerBook Library. Click Select on a bookmark in the "Bookmarks in Library" list to display the annotation for that bookmark in the "Comment for" window. Double-click SELECT on a bookmark to display the corresponding document in the Viewer window

Click SELECT on the "Delete Bookmark" button to delete the currently selected bookmark. This operation cannot be undone.

The bookmark annotation in the "Comment for" window, as well as the bookmark title on the "Comment for" line are editable. Use the standard textedit editing functions to modify them, then click SELECT on the "Save Changes" button.

New Bookmark

Click SELECT on the "New Bookmark..." button to create a bookmark to the current page in the Viewer window. In the resulting New Bookmark pop-up window, edit the bookmark title, and annotate it if desired, then click SELECT on the "Create" button to add the new bookmark to the bookmark list for the current AnswerBook Library. The updated bookmark list is automatically saved to the current Library file so that it can be browsed the next time that file is loaded into **navigator**.

Modify Library

Click SELECT on the "Modify Library..." button to bring up a list of available AnswerBooks to include in the current AnswerBook Library. Select one or more AnswerBooks from the list, then click SELECT on the "Apply" button to put those AnswerBooks in the current Library. This list of AnswerBooks is automatically saved to the current Library file so that it is available when you next run **navigator**.

The AnswerBooks shown in the Modify Library list are those found in the AnswerBook Card Catalogs in your Card Catalog path. See **ab_cardcatalog(4)** for more information.

To prevent two people from updating the same AnswerBook Library file at the same time (and thus losing data and/or corrupting the file), **navigator** sets an advisory lock on the file upon opening it. If the file is already locked, **navigator** gives you the option of opening it read-only, or resetting the lock. If you open an AnswerBook Library file read-only, changes to that Library (including bookmark changes) will not be saved. You should reset a lock only if you know that it is no longer valid (e.g., the navigator that set it is no longer running).

Starting Navigator

You should start **navigator** via the **answerbook(1)** script rather than running it directly. Command line arguments to **answerbook** are passed on to **navigator**.

You can also start AnswerBook by opening an AnswerBook Library file in File Manager, MailTool, or other DeskSet application. See **filemgr(1)** for details.

By default, **navigator** loads your personal AnswerBook Library file (`$HOME/.ab_library`) when it starts. Use the **-b library-file** option to load a different Library file (see **OPTIONS**).

Foreign Language Support

Some AnswerBooks contain translated documents in addition to the English versions. The user can specify the preferred language at the beginning of an AnswerBook session via the `$LANG` environment variable or the "-l" command line flag. **navigator** and **docviewer** will display, search, etc., documents in the preferred language when they are present in the AnswerBook.

ENVIRONMENT**AB_CARDCATALOG**

Specify the name of the card catalog file used to locate AnswerBooks. See **ab_cardcatalog(4)** for more information.

LANG

Specify the preferred language for browsing, searching, etc. Can be overridden by "-l" command line flag.

FULTEMP

Directory for writing temporary files used during search operation.

FILES**~/ab_cardcatalog**

Default AnswerBook Card Catalog file used to locate AnswerBooks. See **ab_cardcatalog(4)** for more information.

~/ab_library

Default AnswerBook Library file loaded by **navigator**. See **ab_library(4)** for more information.

/usr/tmp/ft***\$FULTEMP/ft***

Temporary files used during search operations.

SEE ALSO

docviewer(1), **answerbook(1)**, **ab_admin(1)**, **setlocale(3)**, **ab_cardcatalog(4)**, **ab_library(4)**,

NOTES

AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc.

| | |
|--------------------|---|
| NAME | oclock – display time of day |
| SYNOPSIS | oclock [<i>-options...</i>] |
| DESCRIPTION | Oclock simply displays the current time on an analog display |
| OPTIONS | <p>-fg <i>foreground color</i> choose a different color for the both hands and the jewel of the clock</p> <p>-bg <i>background color</i> choose a different color for the background.</p> <p>-jewel <i>jewel color</i> choose a different color for the jewel on the clock.</p> <p>-minute <i>minute color</i> choose a different color for the minute hand of the clock.</p> <p>-hour <i>hour color</i> choose a different color for the hour hand of the clock.</p> <p>-backing WhenMapped Always NotUseful selects an appropriate level of backing store.</p> <p>-geometry <i>geometry</i> define the initial window geometry; see X11(7).</p> <p>-display <i>display</i> specify the display to use; see X11(7).</p> <p>-bd <i>border color</i> choose a different color for the window border.</p> <p>-bw <i>border width</i> choose a different width for the window border. As the Clock widget changes its border around quite a bit, this is most usefully set to zero.</p> <p>-noshape causes the clock to not reshape itself and ancestors to exactly fit the outline of the clock.</p> <p>-transparent causes the clock to consist only of the jewel, the hands, and the border.</p> |
| COLORS | <p>If you would like your clock to be viewable in color, include the following in the #ifdef COLOR section you read with xrdb:</p> <pre>*customization: -color</pre> <p>This will cause oclock to pick up the colors in the app-defaults color customization file: /usr/openwin/lib/app-defaults/Clock-color. Below are the default colors:</p> <pre>Clock*Background: grey</pre> |

Clock*BorderColor: light blue
Clock*hour: yellow
Clock*jewel: yellow
Clock*minute: yellow

SEE ALSO

X11(7), X Toolkit documentation

COPYRIGHT

Copyright 1989, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Keith Packard, MIT X Consortium

| | |
|--------------------|---|
| NAME | olbiff – OLIT mailbox monitor |
| SYNOPSIS | olbiff [-show <i>n</i>] [-interval <i>n</i>] [-fl <i>n</i>] [-sl <i>n</i>] [-bell] [-help] [-center] [-history] |
| DESCRIPTION | Olbiff monitors the user's mail spool file at a specified interval, and displays information about incoming messages. |
| OPTIONS | -show <i>n</i> display <i>n</i> lines in ListBox -interval <i>n</i> check mail every <i>n</i> seconds -fl <i>n</i> display no more than <i>n</i> characters for name -sl <i>n</i> display no more than <i>n</i> characters in subject -bell beep the bell when each new message arrives -help display the help message -center center the popup on the screen -history maintain the list of items beyond mail increments until the Acknowledge button is pressed |
| SEE ALSO | xbiff(1) |

| | |
|--------------------|--|
| NAME | olitsampler – OLIT widget demo program |
| SYNOPSIS | olitsampler |
| DESCRIPTION | Olitsampler is a simple application program that demonstrates OLIT widgets. |
| SEE ALSO | <i>OLIT Reference Manual</i> |

| | |
|--------------------|--|
| NAME | olittable – OLIT widget demo program |
| SYNOPSIS | olittable |
| DESCRIPTION | <p>Olittable is a simple application program that demonstrates OLIT widgets laid out in an imitation periodic table.</p> <p>This splendid program will execute standalone, but to display additional information about each widget, the program requires a number of auxiliary files, named *.txt. These files contain additional information about widget resources and usage.</p> <p>By default, olittable looks in the directory \$OPENWINHOME/lib/help/olittable, then in the user's current working directory.</p> |
| ENVIRONMENT | TABLEINFO – overrides the location of the *.txt files containing the auxiliary widget information. |
| FILES | *.txt each widget has a file whose base filename is the abbreviated symbol used in the periodic table ("Rb" for RectButton , or "Cp" for Caption .) |
| SEE ALSO | <i>OLIT Reference Manual</i> |

| | |
|----------------------------|---|
| NAME | olmh – OpenLook interface to the MH message handling system |
| SYNOPSIS | olmh [-path <i>mailpath</i>] [-initial <i>foldername</i>] [-flag] [-toolkitoption...] |
| DESCRIPTION | <p>The olmh program provides a window-oriented user interface to the Rand <i>MH</i> Message Handling System. To actually do things with your mail, it makes calls to the <i>MH</i> package. Electronic mail messages may be composed, sent, received, replied to, forwarded, sorted, and stored in folders.</p> <p>To specify an alternate collection of mail folders in which to process mail, use -path followed by the pathname of the alternate mail directory. The default mail path is the value of the Path component in \$HOME/.mh_profile, or \$HOME/Mail if the <i>MH</i> Path is not given. To specify an alternate folder which may receive new mail and is initially opened by olmh, use the -initial flag. The default initial folder is 'inbox'. The option -flag will cause olmh to attempt to change the appearance of its icon when new mail has arrived. These three options have corresponding application-specific resources, named MailPath, InitialFolder, and MailWaitingFlag, which can be used in a resource file. The standard toolkit command line options are given in X11(7).</p> <p>Please don't be misled by the size of this document. It introduces many aspects of the OLIT Widget Set, and provides extensive mechanism for customization of the user interface. olmh really is easy to use.</p> |
| INSTALLATION | <p>The current version of olmh requires that the user is already set up to use <i>MH</i>, version 6. To do so, see if there is a file called .mh_profile in your home directory. If it exists, check to see if it contains a line that starts with "Current-Folder". If it does, you've been using version 4 or earlier of <i>MH</i>; to convert to version 6, you must remove that line. (Failure to do so causes spurious output to stderr, which can hang olmh depending on your setup.)</p> <p>If you do not already have a .mh_profile, you can create one (and everything else you need) by typing "inc" to the shell. You should do this before using olmh to incorporate new mail.</p> <p>For more information, refer to the mh(1) documentation.</p> |
| BASIC SCREEN LAYOUT | <p>olmh starts out with a single window, divided into four main areas:</p> <ul style="list-style-type: none"> – Five buttons with pull-down command menus. – A collection of buttons, one for each top level folder. New users of <i>mh</i> will have two folders, "drafts" and "inbox". – A listing, or Table of Contents, of the messages in the open folder. Initially, this will show the messages in "inbox". |

- A view of one of your messages. Initially this is blank.

OLMH AND THE OLIT WIDGET SET

olmh uses the X Toolkit Intrinsic and the OLIT Widget Set. Many of the features described below (scrollbars, buttonboxes, etc.) are actually part of the OLIT Widget Set, and are described here only for completeness. For more information, see the OLIT Widget Set documentation.

PROCESSING YOUR MAIL

This section will define the concepts of the selected folder, current folder, selected message(s), current message, selected sequence, and current sequence. Each **olmh** command is introduced.

SELECTED FOLDER

A folder contains a collection of mail messages, or is empty.

The selected folder is whichever foldername appears in the bar above the folder buttons. Note that this is not necessarily the same folder that is being viewed. To change the selected folder, just press SELECT on the desired folder button and then SELECT 'Open Folder' from the main 'Folder' menu; if the folder has subfolders, SELECT a folder from the pull down menu. You can traverse to any depth of sub-folders using the pull-right menu buttons.

The Table of Contents, or toc, lists the messages in the viewed folder. The title bar above the Table of Contents displays the name of the viewed folder.

The toc title bar also displays the name of the viewed sequence of messages within the viewed folder. Every folder has an "all" sequence, which contains all the messages in the folder, and initially the toc title bar will show "inbox:all".

FOLDER COMMANDS

The *folder* command menu contains commands of a global nature:

Open Folder

Display the data in the selected folder. Thus, the selected folder also becomes the viewed folder.

Open Folder in New Window

Displays the selected folder in an additional main window. Note, however, that you may not reliably display the same folder in more than one window at a time, although **olmh** will not prevent you from trying.

Create Folder

Create a new folder. You will be prompted for a name for the new folder; to enter the name, move the pointer to the blank box provided and type. Subfolders are created by specifying the parent folder, a slash, and the subfolder name. For example, to create a folder named "olmh" which is a subfolder of an

existing folder named "clients", type "clients/olmh". Click on the Apply button when finished; click on Cancel to cancel this operation.

Delete Folder

Destroy the selected folder. You will be asked to confirm this action (see CONFIRMATION WINDOWS). Destroying a folder will also destroy any subfolders of that folder.

Close Window

Exits **olmh**, after first confirming that you won't lose any changes; or, if selected from any additional **olmh** window, simply closes that window.

**HIGHLIGHTED
MESSAGES,
SELECTED
MESSAGES,
CURRENT
MESSAGE**

It is possible to highlight a message in the area of the Table of Contents. To highlight a message, click on it with pointer button 1.

The selected message is the same as the highlighted message, if any. If no message is highlighted, then the selected message is considered the same as the current message.

The current message is indicated by a '+' next to the message number. It usually corresponds to the message currently being viewed. When a message is viewed, the title bar above the view will identify the message.

**TABLE OF
CONTENTS
COMMANDS**

The *Table of Contents* command menu contains commands which operate on the open, or viewed folder.

Incorporate New Mail

Add any new mail received to your inbox folder, and set the current message to be the first new message. (This command is selectable only if "inbox" is the folder being viewed.)

Commit Changes

Execute all deletions, moves, and copies that have been marked in this folder.

Pack Folder

Renumber the messages in this folder so they start with 1 and increment by 1.

Sort Folder

Sort the messages in this folder in chronological order. As a side effect, this also packs the folder.

Rescan Folder

Rebuild the list of messages. This can be used whenever you suspect that **olmh**'s idea of what messages you have is wrong. (In particular, this is necessary if you change things using straight *MH* commands without using **olmh**.)

Read in Reverse

Read in Reverse will switch the meaning of the next and previous messages, and will increment in the opposite direction. This is useful if you want to read your messages in the order of most recent first. The option acts as a toggle; select it from the menu a second time to

undo the effect.

MESSAGE COMMANDS

The *Message* command menu contains commands which operate on the selected message(s), or if there are no selected messages, the current message.

Compose Message Composes a new message. A new window will be brought up for composition; a description of it is given in the COMPOSITION WINDOWS section below. This command does not affect the current message.

View Next Message

View the first selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message after the current message.

View Previous

View the last selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message before the current message.

Mark Deleted

Mark the selected messages for deletion. If no messages are highlighted, then this will mark the current message for deletion and automatically display the next unmarked message.

Mark Move

Mark the selected messages to be moved into the current (selected) folder. (If the current folder is the same as the viewed folder, this command will just beep.) If no messages are highlighted, this will mark the current message to be moved and display the next unmarked message.

Mark Copy

Mark the selected messages to be copied into the current folder. (If the current folder is the same as the viewed folder, this command will just beep.) If no messages are highlighted, mark the current message to be copied.

Unmark

Remove any of the above three marks from the selected messages, or the current message, if none are highlighted.

View in New Window

Create a new window containing only a view of the first selected message, or the current message, if none are highlighted.

Reply

Create a composition window in reply to the first selected message, or the current message, if none are highlighted.

Forward

Create a composition window whose body is initialized to be the contents of the selected messages, or the current message if none are highlighted.

Use as Composition

Create a composition window whose body is initialized to be the contents of the first selected message, or the current message if none

are selected. Any changes you make in the composition will be saved in a new message in the “drafts” folder, and will not change the original message. However, this command was designed to be used within the “drafts” folder to compose message drafts, and there is an exception to this rule. If the message to be used as composition was selected from the “drafts” folder, the changes will be reflected in the original message (see COMPOSITION WINDOWS).

Print

Print the selected messages, or the current message if none are selected. **olmh** normally prints by invoking the **enscript(1)** command, but this can be customized with the application-specific resource **PrintCommand**.

**SEQUENCE
COMMANDS**

The *Sequence* command menu of xmh allows the user to view chosen sequences of messages. Unfortunately this useful feature is not available in this demonstration version of Olmh.

VIEW COMMANDS

Commands in the View menu and in the buttonboxes of view windows (which result from the Message command “View In New”) correspond in functionality to commands of the same name in the Message menu, but they operate on the viewed message rather than the selected messages or current message.

Close Window

When the viewed message is in a separate view window, this command will close the view, after confirming the status of any unsaved edits.

Reply

Create a composition window in reply to the viewed message.

Forward

Create a composition window whose body is initialized to be the contents of the viewed message.

Use As Composition

Create a composition window whose body is initialized to be the contents of the viewed message. Any changes made in the composition window will be saved in a new message in the “drafts” folder, and will not change the original message. An exception: if the viewed message was selected from the “drafts” folder, the original message is edited.

Edit Message

This command enables the direct editing of the viewed message.

Save Message

This command is insensitive until the message has been edited; when activated, edits will be saved to the original message in the view.

Print

Print the viewed message. **olmh** prints by invoking the **enscript(1)** command, but this can be customized with the application-specific resource **PrintCommand**.

**COMPOSITION
WINDOWS**

Aside from the normal text editing functions, there are six command buttons associated with composition windows:

| | |
|------------------------|---|
| Close Window | Close this composition window. If changes have been made since the most recent Save or Send, you will be asked to confirm losing them. |
| Send | Send this composition. |
| New Headers | Replace the current composition with an empty message. If changes have been made since the most recent Send or Save, you will be asked to confirm losing them. |
| Compose Message | Bring up another new composition window. |
| Save Message | Save this composition in your drafts folder. Then you can safely close the composition. At some future date, you can continue working on the composition by opening the drafts folder, selecting the message, and using the "Use as Composition" command. |
| Insert | Insert a related message into the composition. If the composition window was created with a "Reply" command, the related message is the message being replied to, otherwise no related message is defined and this button is insensitive. The message may be filtered before being inserted; see ReplyInsertFilter under APPLICATION RESOURCES for more information. |

ACCELERATORS

Accelerators are shortcuts. They allow you to invoke commands without using the menus, directly from the keyboard.

olmh defines the following keyboard accelerators over the surface of the main window, except in the view area while editing a message:

| | |
|------------|--|
| Meta-I | Incorporate New Mail |
| Meta-C | Commit Changes |
| Meta-R | Rescan Folder |
| Meta-P | Pack Folder |
| Meta-S | Sort Folder |
| Meta-space | View Next Message |
| Meta-c | Mark Copy |
| Meta-d | Mark Deleted |
| Meta-f | Forward the selected or current message |
| Meta-m | Mark Move |
| Meta-n | View Next Message |
| Meta-p | View Previous Message |
| Meta-r | Reply to the selected or current message |
| Meta-u | Unmark |

In addition to these keyboard accelerators, you can use all the standard OLIT mouseless commands to traverse and activate objects. For example, to scroll a TextEdit widget inside a ScrollingList:

```

Alt-Down      Scroll down a line
Alt-Up        Scroll up a line
Alt-PgDn      Scroll down a page
Alt-PgUp      Scroll up a page
Alt-Ctrl-End  Scroll to bottom
Alt-Ctrl-Home Scroll to top
    
```

**TEXT EDITING
COMMANDS**

All of the text editing commands are actually defined by the TextEdit widget in the OLIT Widget Set. The commands may be bound to different keys than the defaults through the OLIT key re-binding mechanisms. See the OLIT Widget Set documentation for more details.

**CONFIRMATION
DIALOG BOXES**

Whenever you press a button that may cause you to lose some work or is otherwise dangerous, a popup dialog box will appear asking you to confirm the action. This window will contain an "Abort" or "No" button and a "Confirm" or "Apply" button. Pressing the "No" button cancels the operation, and pressing the "Apply" will proceed with the operation.

**WIDGET
HIERARCHY**

In order to specify resources, it is useful to know the hierarchy of widgets which compose **olmh**. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name. The application class name is Olmh.

The hierarchy of the main toc and view window is identical for additional toc and view windows, except that a TopLevelShell widget is inserted in the hierarchy between the application shell and the RubberTile widget.

```

Olmh olmh
  RubberTile olmh
    Control menuBox
      MenuButton folderButton
        MenuShell menu
          Form menu_form
            Control pane
              OblongButton open
              OblongButton openInNew
              OblongButton create
              OblongButton delete
              OblongButton close
    MenuButton tocButton
    
```

```
MenuShell menu
  Form menu_form
    Control pane
      OblongButton inc
      OblongButton commit
      OblongButton pack
      OblongButton sort
      OblongButton rescan
      OblongButton reverse
MenuButton messageButton
  MenuShell menu
    Form menu_form
      Control pane
        OblongButton compose
        OblongButton next
        OblongButton prev
        OblongButton delete
        OblongButton move
        OblongButton copy
        OblongButton unmark
        OblongButton viewNew
        OblongButton reply
        OblongButton forward
        OblongButton useAsComp
        OblongButton print
MenuButton sequenceButton
  MenuShell menu
    Form menu_form
      Control pane
        OblongButton pick
        OblongButton openSeq
        OblongButton addToSeq
        OblongButton removeFromSeq
        OblongButton deleteSeq
        OblongButton all
MenuButton viewButton
  MenuShell menu
    Form menu_form
      Control pane
        OblongButton reply
        OblongButton forward
        OblongButton useAsComp
        OblongButton edit
        OblongButton save
        OblongButton print
```

```

StaticText folderTitlebar
Control folders
  OblongButton inbox
  OblongButton drafts
  OblongButton a_folder
  MenuButton sub_folder
    MenuShell menu
      Form menu_form
        Control pane
          OblongButton sub_folder
          OblongButton sub_folder/this
          OblongButton sub_folder/that
          .
          .
          .
    
```

```

StaticText tocTitlebar
ScrolledWindow scrwin
  TextEdit toc
StaticText viewTitlebar
ScrolledWindow scrwin
  TextEdit view
    
```

The hierarchy of the Create Folder popup dialog box:

```

PopupWindowShell prompt
  FooterPanel panel
    Control control
      Control upper
        StaticText label
        TextField prompt
      Control lower
        OblongButton apply
        OblongButton cancel
    
```

The hierarchy of the Notice dialog box, which reports messages from MH:

```

NoticeShell notice
  Control pane
    StaticText textarea
    Control controlarea
      OblongButton confirm
    
```

The hierarchy of the Confirmation dialog box:

```

NoticeShell confirm
  Control pane
    StaticText textarea
    Control controlarea
      OblongButton yes
      OblongButton no

```

The hierarchy of the dialog box which reports errors:

```

NoticeShell error
  Control pane
    StaticText textarea
    Control controlarea
      OblongButton OK

```

The hierarchy of the composition window:

```

TopLevelShell olmh
  RubberTile olmh
    StaticText composeTitlebar
    ScrolledWindow scrwin
      TextEdit comp
    Control compButtons
      OblongButton close
      OblongButton send
      OblongButton reset
      OblongButton compose
      OblongButton save
      OblongButton insert

```

The hierarchy of the view window:

```

TopLevelShell olmh
  RubberTile olmh
    StaticText viewTitlebar
    ScrolledWindow scrwin
      TextEdit view
    Control viewButtons
      OblongButton close
      OblongButton reply
      OblongButton forward
      OblongButton useAsComp
      OblongButton edit
      OblongButton save
      OblongButton print

```

The hierarchy of the pick window:

Not implemented in this demo version of Olmh

APPLICATION-SPECIFIC RESOURCES

Resource instance names begin with a lower case letter but are otherwise identical to the class name.

If TocGeometry, ViewGeometry, CompGeometry, or PickGeometry are not specified, then the value of Geometry is used instead. If the resulting height is not specified (e.g., "", "=500", "+0-0"), then the default height of windows is calculated from fonts and line counts. If the width is not specified (e.g., "", "=x300", "-0+0"), then half of the display width is used. If unspecified, the height of a pick window defaults to half the height of the display.

Any of these options may also be specified on the command line by using the X Toolkit Intrinsic resource specification mechanism. Thus, to run **olmh** showing all message headers,

```
% olmh -xrm '*HideBoringHeaders:off'
```

The following resources are defined:

Banner A short string that is the default label of the folder, Table of Contents, and view. The default is "olmh OpenWindows V3.0"

BlockEventsOnBusy

Whether to disallow user input and show a busy cursor while **olmh** is busy processing a command. Default is true.

BusyCursor

The name of the symbol used to represent the position of the pointer, displayed if **BlockEventsOnBusy** is true, when **olmh** is processing a time-consuming command. The default is "watch".

BusyPointerColor

The foreground color of the busy cursor. Default is XtDefaultForeground.

CheckFrequency

How often to check for new mail, make checkpoints, and rescan the Table of Contents, in minutes. If **CheckNewMail** is true, **olmh** checks to see if you have new mail each interval. If **MakeCheckpoints** is true, checkpoints are made every fifth interval. Also every fifth interval, the Table of Contents is checked for inconsistencies with the file system, and rescanned. To prevent all of these checks from occurring, set **CheckFrequency** to 0. The default is 1.

CheckNewMail

If true, **olmh** will check at regular intervals to see if new mail has arrived for any of the folders. A visual indication will be given if new mail is waiting to be retrieved. Default is True. (See BUGS). The interval can be adjusted with the

CheckFrequency.**CommandButtonCount**

The number of command buttons to create in a button box in between the toc and the view areas of the main window. **olmh** will create these buttons with the names *button1*, *button2* and so on, in a box with the name *commandBox*. The user can specify labels and actions for the buttons in a private resource file; see the section on Actions. The default is 0.

CompGeometry

Initial geometry for windows containing compositions.

Cursor The name of the symbol used to represent the pointer. Default is "left_ptr".

DraftsFolder

The folder used for message drafts. Default is "drafts".

Geometry

Default geometry to use. Default is none.

HideBoringHeaders

If "on", then **olmh** will attempt to skip uninteresting header lines within messages by scrolling them off. Default is "on".

InitialFolder

Which folder to display on startup. May also be set with the command-line option **-initial**. Default is "inbox".

InitialIncFile

The file name of your incoming mail drop. **olmh** tries to construct a filename for the "inc -file" command, but in some installations (e.g. those using the Post Office Protocol) no file is appropriate. In this case, **InitialIncFile** should be specified as the empty string, and *inc* will be invoked without a -file argument. The default is to use the value of the environment variable **MAIL**, or if that is not set, to append the value of the environment variable **USER** to **/var/mail/**.

MailPath

The full path prefix for locating your mail folders. May also be set with the command-line option, **-path**. The default is the Path component in **\$HOME/.mh_profile**, or "**\$HOME/Mail**" if none.

MailWaitingFlag

If true, **olmh** will attempt to set an indication in its icon when new mail is waiting to be retrieved. If this option is true, then **CheckNewMail** is assumed to be true as well. The **-flag** command line option is a quick way to turn MailWaitingFlag on.

MakeCheckpoints

If true, **olmh** will attempt to save checkpoints of volatile information. The frequency of checkpointing is controlled by the resource **CheckFrequency**.

MhPath What directory in which to find the *MH* commands. If a command isn't found here, then the directories in the user's path are searched. Default is **"/usr/local/mh6"**.

PickGeometry

Initial geometry for pick windows.

PointerColor

The foreground color of the pointer. Default is XtDefaultForeground.

PrefixWmAndIconName

Whether to prefix the window and icon name with "olmh: ". Default is true.

PrintCommand

What sh command to execute to print a message. Note that stdout and stderr must be specifically redirected! If a message or range of messages is selected for printing, the full file paths of each message file is appended to the specified print command. The default is "enscript >/dev/null 2>/dev/null".

ReplyInsertFilter

A shell command to be executed when the *Insert* button is activated in a composition window. The full path and filename of the source message is added to the end of the command before being passed to **sh**(1). The default filter is *cat*; i.e. it inserts the entire message into the composition. Interesting filters are: *awk -e '{print " " \$0}'* or *<mh directory>/lib/mhl -form mhl.body*.

ReverseReadOrder

When true, the next message will be the message prior to the current message in the table of contents, and the previous message will be the message after the current message in the table of contents. The default is false.

SendBreakWidth

When a message is sent from **olmh**, lines longer than this value will be split into multiple lines, each of which is no longer than **SendWidth**. This value may be overridden for a single message by inserting an additional line in the message header of the form *SendBreakWidth: value*. This line will be removed from the header before the message is sent. The default is 85.

SendWidth

When a message is sent from **olmh**, lines longer than **SendBreakWidth** characters will be split into multiple lines, each of which is no longer than this value. This value may be overridden for a single message by inserting an additional line in the message header of the form *SendWidth: value*. This line will be removed from the header before the message is sent. The default is 72.

SkipCopied

Whether to skip over messages marked for copying when using "View Next Message" and "View Previous Message". Default is true.

SkipDeleted

Whether to skip over messages marked for deletion when using "View Next Message" and "View Previous Message". Default is true.

SkipMoved

Whether to skip over messages marked for moving to other folders when using "View Next Message" and "View Previous Message". Default is true.

StickyMenu

If true, when popup command menus are used, the most recently selected entry will be under the cursor when the menu pops up. Default is false. See the file *clients/olmh/Olmh.sample* for an example of how to specify resources for popup command menus.

TempDir

Directory for **olmh** to store temporary directories. For privacy, a user might want to change this to a private directory. Default is “/tmp”.

TocGeometry

Initial geometry for master **olmh** windows.

TocPercentage

In *xmh*, this represents the percentage of the main window that is used to display the Table of Contents. However, in *olmh* this now refers to the relative proportion of a size CHANGE to the main window that gets applied to the toc. Consider the toc and the view portions of the main window as having a weight; The view has a fixed weight of 50 and the toc has a weight of TocPercentage (default is 33). The total weight is 83 and, by default the toc will get 33/83 of the size change and the view will get 50/83 of the size change. To change the INITIAL number of lines in the toc, set the resource *toc.linesVisible to some number. NOTE however that you may not get exactly this number of lines because the RubberTile may resize the toc widget.

TocWidth

How many characters to generate for each message in a folder's table of contents. Default is 100. Use 80 if you plan to use *mhl* a lot, because it will be faster, and the extra 20 characters may not be useful.

ViewGeometry

Initial geometry for windows showing only a view of a message.

**CUSTOMIZATION
USING MH**

The initial text displayed in a composition window is generated by executing the corresponding *MH* command; i.e. *comp*, *repl*, or *forw*, and therefore message components may be customized as specified for those commands. *Comp* is executed only once per invocation of **olmh** and the message template is re-used for each successive new composition.

FILES

~/Mail
 ~/.mh_profile - *MH* profile
 /usr/local/mh6 - *MH* commands
 ~/Mail/<folder>/.olmhcache - scan folder
 ~/Mail/<folder>/.mh_sequences - sequence definitions
 /tmp - temporary files

| | |
|------------------|---|
| SEE ALSO | xrdb(1) , X Toolkit Intrinsics, OLIT Widget Set, mh(1) , enscript(1) |
| BUGS | <ul style="list-style-type: none"> - Marks disappear after Incorporating new mail, although internally the messages are still marked. - Reading mail using 'raw' MH commands or another mailer such as xmh, can cause olmh to get confused, and the 'current' message in the toc may not be the actual message displayed. - Mouseless does not activate buttons in comp or view TopLevel shells. - Because of the way OLIT uses the translation manager to map all events to OlAction, the action procs do not work as for xmh and have been removed from this man page. - Printing support is minimal. - Should handle the "unseen" message-sequence. - Should determine by itself if the user hasn't used <i>MH</i> before, and offer to create the .mh_profile, instead of hanging on inc. - Still a few commands missing (rename folder, remail message). - A bug in <i>MH</i> limits the the number of characters in .mh_sequences to BUFSIZ. When the limit is reached, the .mh_sequences file often becomes corrupted, and sequence definitions may be lost. - Except for the icon, there isn't an indication that you have new mail. - There should be a resource, ShowOnInc, which when true, would show the current message in the view after incorporating new mail. - The CheckFrequency resource should be split into two separate resources. - WM_SAVE_YOURSELF protocol is ignored. - WM_DELETE_WINDOW protocol doesn't work right when requesting deletion of the first toc and view, while trying to keep other olmh windows around. - Doesn't support annotations when replying to messages. - If the MH commands aren't in your PATH, you get "Execvp Failed" messages instead of something useful like "Can't find program." |
| COPYRIGHT | Copyright 1988, 1989, Digital Equipment Corporation. Copyright 1989, Massachusetts Institute of Technology |
| AUTHOR | Terry Weissman, Digital Western Research Laboratory modified by Donna Converse, MIT X Consortium ported to OLIT as a demo by John S Cooper, Sun Microsystems, Inc. |

| | |
|--------------------|--|
| NAME | olwm – OPEN LOOK window manager for OpenWindows |
| SYNOPSIS | olwm [<i>options</i>] |
| DESCRIPTION | Olwm is a window manager for the X Window System that implements parts of the OPEN LOOK graphical user interface. It is the standard window manager for Sun's OpenWindows product, but it will work properly with any X11 system. The only requirements for running olwm are that the server have the OPEN LOOK glyph and cursor fonts available. |
| OPTIONS | <p>Most command-line options have counterparts in the resource database. A command-line option will override any setting from the resource database.</p> <p>-2d Use two-dimensional look. This is the default for monochrome systems.</p> <p>-3d Use three-dimensional look. This is the default for color systems. This option is ignored for monochrome systems.</p> <p>-bd <i>color</i>, -bordercolor <i>color</i> Specifies the border color. See the description of the BorderColor resource.</p> <p>-bg <i>color</i>, -background <i>color</i> Specifies the background color. See the description of the Background resource.</p> <p>-c, -click Use click-to-focus mode. This is the default focus mode.</p> <p>-depth <i>depth</i> Specifies the depth of the visual in which olwm is to run. See the discussion in the Screen Resources section for further information about depths.</p> <p>-display <i>display-string</i> Specify the name of the display to manage. Overrides the DISPLAY environment variable, if any. In addition, the display string is exported to olwm's environment, so processes forked from olwm will inherit this value.</p> <p>-dsdm Specify that olwm should provide the Drop Site Database Management (DSDM) service. This is the default.</p> <p>-f, -follow Use focus-follows-mouse mode. Default mode is click-to-focus.</p> <p>-fn <i>font-name</i>, -font <i>font-name</i> Set the font for window titles.</p> <p>-fg <i>color</i>, -foreground <i>color</i> Specifies the foreground color. See the description of the ForegroundColor resource.</p> <p>-multi Manage windows on all screens that a display supports. This is the default.</p> <p>-name <i>resource-name</i> Use <i>resource-name</i> to look up resources in the resource database.</p> <p>-nodsdm</p> |

Specify that **olwm** should not provide the Drop Site Database Management service. The default is to provide the service.

-single

Manage windows for a single screen only, using the default screen for the specified display. Overrides the **-multi** option.

-syncpid *process-id*

When **olwm** has completed its initialization, it will send a signal (SIGALRM by default) to *process-id*. The signal will be sent only if this option is present. This is useful for running **olwm** from shell scripts (such as **.xinitrc**) in such a way that the script waits for **olwm** to finish its initialization, while leaving **olwm** as a child process of the shell script. This can be done using the following **sh**(1) construct:

```
sleep 15 & pid=$!
olwm -syncpid $pid &
wait $pid
```

-syncsignal *signal*

Specifies the signal to send instead of SIGALRM. The signal is specified as a number, not symbolically.

-visual *visual-class*

Specifies the class of the visual in which **olwm** is to run. See the discussion in the Screen Resources section for further information about visuals.

-xrm *resource-string*

Specify resources on the command-line. Resources specified here will override resources found in resource files.

**DEBUGGING
OPTIONS**

The following options are strictly for debugging. They are not recommended for general use. Don't use them unless you know what you are doing.

-all Print a message for every event received.

-debug

Equivalent to turning on all debugging options.

-orphans

Print orphaned events. Orphaned events are events that are associated with a window or frame that has no entry in the frame hash table, or events that are not handled by the various event handlers.

-synchronize

Run the window manager in synchronous mode.

**INTERNATIONAL-
IZATION
OPTIONS**

-basiclocale *locale-name*

Specifies the basic OPEN LOOK locale category setting. This category will be the base for other locale categories.

-displaylang *locale-name*

Specifies the display language OPEN LOOK locale category. This category

**LOCALE
HANDLING**

affects the contents of workspace menu, window menu and notice messages.

-numeric *locale-name*

Specifies the numeric format OPEN LOOK locale category. This category affects the numeric format displayed in any message that contains numerics.

The *locale* is the set of language and cultural conventions used by a program. The locale controls the language-dependent part of **olwm**'s behavior. The OPEN LOOK international extensions have defined several locale categories as follows:

Basic Locale

This is the basic setting for the entire locale mechanism. This category specifies internal character handling behavior.

Display Language

This category specifies the language used for displaying menus, notice messages, and error messages.

Input Language

This category specifies the language used for text input. This category has no effect on **olwm**, because it does not accept text input from the keyboard.

Date Format

This category specifies the format of date and time. This category has no effect on **olwm**, because it does not display any date and time information.

Numeric Format

This category specifies the format of displayed numeric data.

The Basic Locale setting determines the character set used by **olwm**. The other locale categories can differ from the basic setting, but they cannot require a different character set from the Basic Locale. The following restrictions thus apply:

1. If basic locale setting is the "C" locale, then all other locale categories must be in the "C" locale.
2. If the Basic Locale is set to a locale other than the "C" locale, then all other locale categories must be set either to a locale that uses the same character set as the basic setting, or to the "C" locale.

The following methods are available to inform **olwm** of the locale settings, listed in order of priority:

1. Command line options (such as **-basiclocale**);
2. by resource database; and
3. **setlocale(3C)** function defaults (e.g. LANG environment variable).

INPUT FOCUS

The *input focus* is the window that will receive keystrokes. **olwm** has two different input focus modes, which are different ways of transferring the input focus from one window to another. By default, **olwm** uses "click-to-focus" (also known as "click-to-type") mode. This means that you must click on the window in order to get the focus to it. While a window has the input focus, the pointer can be anywhere on the screen; the keyboard events will still go to that window. You can set the input focus to a window and

simultaneously raise it to the top by clicking the left mouse button in the window's title bar or border.

olwm has another focus mode called "focus-follows-mouse." In this mode, whatever window the mouse is pointing to will receive the input focus. To switch the input focus from one window to another, you simply move the pointer to the other window; you don't have to click at all. Note, however, that to transfer the focus amongst subwindows of a single top-level window, you must click in the subwindow, or you must use focus transfer function keys (if available from the application).

The input focus mode can be controlled with command-line options or by entries in the resource database. Neither focus mode has inherent advantages. Which one you choose is a matter of personal preference.

MOUSE BUTTONS

OPEN LOOK defines three mouse button functions: SELECT, ADJUST, and MENU. On systems with three mouse buttons, these functions are mapped to buttons 1, 2, and 3 (left, middle, and right) respectively. On systems with two mouse buttons, SELECT is on button 1 (left) and MENU is on button 2 (right). ADJUST can be performed by holding down the Shift key while pressing button 1. On systems with a single mouse button, that button is SELECT. Holding Shift while pressing the button gives ADJUST, and holding Control gives MENU.

There is an alternate style of button handling for two-button mice: SELECT is button 1, ADJUST is button 2, and MENU is performed by holding down buttons 1 and 2 simultaneously. This technique is referred to as mouse button *chording*. Turns on the mouse chording mechanism. This allows two-button mice to have a different button binding than the OPEN LOOK defaults for mice with fewer than three buttons. When **OpenWindows.MouseChordMenu** is set to True, mouse button 1 is SELECT, mouse button 2 is ADJUST, and pressing the SELECT and the ADJUST buttons at the same time will act as the MENU button. For the default behavior when this resource is set to False, see the section "MOUSE BUTTONS" in the **olwm(1)** manual page.

MANIPULATING WINDOWS AND ICONS

Window Title Bar and Borders.

Clicking SELECT selects the window, raises it above other windows, and deselects any other objects. In click-focus mode, the focus is also transferred to this window. Pressing and holding SELECT and then dragging the mouse will move windows without raising them or setting the focus. If this window is selected, it and all other selected windows are moved simultaneously. Otherwise, just this window is moved, and it is not selected. If you hold down the Control key while you are moving a window, motion is constrained either vertically or horizontally, depending on which direction you move first.

Double-clicking SELECT on the window is the same as selecting the Full Size (or Restore Size) menu item. Clicking ADJUST will toggle the selected state of this window. If other windows or icons are already selected, they remain selected. ADJUST is useful for selecting several windows and icons. Pressing MENU will bring up the window menu. See the Window Menu section for further details. If the **Alt** key is held down, the mouse button functions become accessible anywhere over the window, not just over the title bar and borders. The modifier used can be changed; see the description of the **WMGrab**

resource in the section on Modifier Customization.

Resize Corners.

You can resize a window by pressing and holding SELECT over any of the resize corners and then dragging the mouse to the new location. Releasing the mouse button will set the new size of the window. If you hold down the Control key while you are dragging, the resize operation is constrained to resize vertically or horizontally, depending on which direction you move first.

Window Button.

The Window Button is the small box with a downward-pointing triangle near the left end of the title bar. Pressing MENU over the window button will bring up the Window Menu. Clicking SELECT over the left mouse button on the Window Button will execute the window menu's default action. This will usually close the window into an icon. You can change the window menu's default action by holding down the Control key while manipulating the window menu.

Pushpin.

OPEN LOOK pop-up windows have a pushpin instead of a window button. The pin is either in or out, and you can click SELECT on the pin to move it to the other state. If the pin is out, pressing a command button inside the window will execute the command and then dismiss (take down) the window. If the pin is in, the window is "pinned" to the workspace, and it will remain on the screen even after you have pressed a command button in the window. This allows you to press several command buttons in the same window. Pulling the pin out (by clicking SELECT over it) will dismiss the window immediately.

Icons.

An icon represents a closed window. You can still do most of the same operations as with an open window. Moving and selecting icons with SELECT and ADJUST is exactly the same as for open windows. A similar version of the Window Menu is available on an icon by pressing MENU. Double-clicking SELECT will open the icon. Icons cannot be resized.

NON-RECTANGULAR WINDOWS

The X11 Non-Rectangular Window Shape Extension (commonly referred to simply as the SHAPE extension) allows windows to have arbitrary shapes. **Olwm** will handle these windows by giving them no decoration whatsoever. Shaped windows can be manipulated by using the WMGrab modifier (Alt by default) with the mouse buttons. (See the section on Modifier Customization for further details.) Shaped windows can be moved, resized, closed, opened, etc. like ordinary windows. The selection feedback for shaped windows is the presence of resize corners floating at the corners of the bounding rectangle of the window's shape.

SELECTIONS ON THE WORKSPACE

You can select a group of windows and icons by using the left or middle mouse buttons over the Workspace (the area of the screen outside of all windows and icons, commonly known as the "root window"). Pressing either SELECT or ADJUST and dragging the mouse will define a rubber-band rectangle. When you release the mouse button, the set

**MENU
OPERATION**

of windows and icons enclosed by this rectangle will be operated on. If you created the rectangle using SELECT, the windows and icons within will be selected, and all other objects will be deselected. If you used ADJUST, the objects within will have their selected state toggled, and any other windows and icons already selected will remain selected.

In general, pop-up menus are operated using the MENU mouse button. There are two methods of operating with an OPEN LOOK menu: the "click-move-click" method and the "press-drag-release" method. You choose the method either by clicking the MENU button (pressing and releasing it quickly) or by pressing it down and holding it. If you click the MENU button, the menu will pop up and will stay up indefinitely. To continue operating the menu, click the MENU button over a menu item. To dismiss the menu, click the MENU button on an area of the screen outside the menu. To operate menus in press-drag-release mode, press the MENU button and hold it down while you move the mouse. The menu will remain on the screen as long as you hold down the MENU button. To execute an action, move the pointer over a menu item and release the mouse button. To dismiss the menu, move the pointer outside the menu and release the MENU button.

Some menu items have a sub-menu. This is indicated by a right-pointing triangle at the right edge of the item. To activate a submenu, click on the item (in click-move-click mode) or move the pointer to the item and then move toward the right edge of the menu (in press-drag-release mode).

Some menus have pushpins. If a menu has a pushpin, it will initially be in the "out" state. If you click on the pin (in click-move-click mode) or move over it and release (in press-drag-release mode) you will pin the menu to the workspace. The menu will remain on the screen indefinitely and you can execute commands from it by clicking on its items. To remove the menu, move over the pin and click SELECT on it.

The behavior of menus can be customized using olwm's resources. In the Global Resources section, see the entries for **ClickMoveThreshold**, **DragRightDistance**, **MultiClickTimeout**, and **SelectDisplaysMenu** for further information.

Some menus may have "accelerators" defined for them. See the section on Menu Accelerators for further details.

**WORKSPACE
MENU**

Pressing MENU over the workspace brings up the Workspace Menu. This menu is customizable, but it typically contains at least the following items. (The items may appear in a different language depending on the current locale setting.)

Programs

This item has a sub-menu that allows you to invoke applications. The default Programs sub-menu contains all of the programs in the OpenWindows DeskSet. However, users typically customize this menu to contain many more programs and to contain nested sub-menus. See the section on Menu Customization for further information.

Utilities

This item has a sub-menu that contains several utility functions for the workspace, including Refresh (redisplay all windows on the screen), Lock Screen, and Save Workspace.

WINDOW MENU**Properties...**

This item brings up the Workspace Properties window, which allows you to view and customize settings of the OpenWindows environment.

Help... Brings up the table of contents of the Help Handbooks.

Desktop Intro...

Brings up a tutorial introduction to the Sun Desktop.

Exit Shuts down all applications and exits the window system. A confirmation notice will appear first to give you a chance to cancel the operation.

The window menu of most windows has the following items. (The items may appear in a different language depending on the current locale setting.)

Close Closes the window to an icon. Any OPEN LOOK pop-up windows are closed into this icon as well. They will reappear when the icon is opened. This item is "Open" if you bring up this menu over an icon.

Full Size

Expands the window to the full height of the screen. If this has already done, the button is Normal Size instead of Full Size. Normal Size restores the window to the size it was before you did the Full Size operation. If the application has specified a maximum size for the window, this size is used for Full Size instead of the full screen height.

Move Starts the keyboard-based form of moving the window. Appears only if OPEN LOOK Mouseless Mode is enabled.

Resize Starts the keyboard-based form of resizing the window. Appears only if OPEN LOOK Mouseless Mode is enabled.

Back Moves the window behind all other windows.

Refresh

Clears and redisplay the window.

Quit Kills the program running in the window and removes the window. If the application has elected to participate in the WM_DELETE_WINDOW protocol, **olwm** sends a WM_DELETE_WINDOW ClientMessage instead of killing that window.

OPEN LOOK pop-up windows (as opposed to base windows) have a smaller window menu. It lacks the Close, Full Size, and Quit items, but it has two new items:

Dismiss

Causes the window to be dismissed. This button has a sub-menu with two items: This Window, which dismisses just this window, and All Pop-ups, which dismisses all pop-up windows owned by this application.

Owner?

Raises and flashes the title bar of the base window that "owns" this pop-up window.

MENU CUSTOMIZATION FILES

You can customize **olwm**'s Workspace Menu by putting a menu description into a file that **olwm** will read. When it starts up, **olwm** will first look for a file named by the `OLWMMENU` environment variable. If this variable does not exist, or if the file is not readable, **olwm** will then look in the file named `.openwin-menu` in your home directory. If this file is not present or is unreadable, **olwm** will fall back on the system default menu file. If, for some reason, the system default menu file cannot be found, **olwm** will use a minimal, built-in menu. The menu file that is read can also be modified by the display language locale setting. The locale name is used as a suffix for the filename. If a localized menu file is found, it is used in preference to the non-localized menu file. For example, if the display language locale is "japanese", the file `.openwin-menu.japanese` will take precedence over the file `.openwin-menu`.

Olwm will automatically re-read its menu file whenever the menu file changes. This lets you make many small changes to a menu file, trying out the modified menu after each change. The automatic re-reading can be controlled with the **AutoReReadMenuFile** resource.

If **olwm** encounters a syntax error during the reading of any menu file, a message is printed to the standard error file and the reading of this menu file is considered to have failed. **Olwm** will then attempt to read the next file in the sequence as described above.

MENU SPECIFICATION SYNTAX

The menu specification language has a number of keywords, all of which are in all upper case letters. The keywords are *not* translated into the language specified by the the locale category settings. Keywords are always in English.

Each line typically specifies one menu button. There are three fields on each line: a label, the optional keyword `DEFAULT`, and a command. The label is either a single word or a string enclosed in double quotes. This is the label that appears in the menu button. If the optional keyword `DEFAULT` appears next, this menu item becomes the default item for this menu. The rest of the line (excluding leading whitespace) is considered to be a command. It is executed by sending it to `sh(1)`. Any shell metacharacters will be passed through to the shell unchanged. The command field can be extended onto the next line by placing a backslash `'\'` at the end of the line. The newline will not be embedded in the command.

A sub-menu is specified using the special keyword `MENU` in place of a command. A button is added to the current menu, and clicking or pulling right on this button will bring up the sub-menu. Subsequent lines in the menu file define buttons for the sub-menu, until a line that has the special keyword `END` in the command field is encountered. The label of the `MENU` line must match the label on the `END` line, otherwise an error is signaled. Sub-menus can be nested arbitrarily, bracketed by `MENU` and `END` lines with matching labels.

Sub-menus can be defined in a different file using either the `MENU` or the `INCLUDE` keyword. To include a sub-menu from another file, use a line with a label, either the `MENU` or the `INCLUDE` keyword, and then the filename. The file so named is assumed to contain lines that specify menu buttons. The sub-menu file need not have any `MENU` or `END` lines (unless it has sub-menus itself). The current file need not have a matching `END` line if the sub-menu is read from another file. Sub-menu files included with the

MENU keyword are considered to be an integral part of the menu tree, and any error encountered during reading of the file will cause the entire menu to be considered invalid. A sub-menu file included with the INCLUDE keyword is considered optional, and any error encountered during reading of the file is not considered fatal. If an error occurs during INCLUDE processing, a disabled (grayed-out) item is inserted in place of the sub-menu and processing of the current menu file continues.

To make a sub-menu pinnable, add the special keyword "PIN" after the END keyword on the line that ends the sub-menu definition, or after the TITLE directive (see below).

By default, the label in a menu button is used as the title of the sub-menu. This can be overridden by specifying a line that has the special keyword TITLE in the command field. The label from this line will be used as the sub-menu's title. This line can appear anywhere in the sub-menu definition. It does not add an item to the menu. In addition, if the PIN keyword follows the TITLE keyword on this line, the sub-menu will be made pinnable. This construct is useful for declaring that a sub-menu defined in a separate file be pinnable.

A line containing only the keyword SEPARATOR will add extra space before the next item.

The following keywords can be used in the command field of a menu item. They specify functions that are internal to **olwm**, that are not invoked by running a shell.

BACK_SELN

Move the selected windows and icons behind other windows.

EXIT

Kills all applications and exits the window manager after getting confirmation from the user. This is useful for exiting the entire window system.

EXIT_NO_CONFIRM

Like EXIT but skips the confirmation notice.

FLIPDRAG

Toggle the state of the **DragWindow** resource.

FLIPFOCUS

Toggle the state of the **SetInput** resource.

FULL_RESTORE_SIZE_SELN

Toggle the full-sized/normal-sized states of the selected windows and icons.

NOP

No operation; don't do anything.

OPEN_CLOSE_SELN

Toggle the opened/closed states of the selected windows and icons.

QUIT_SELN

Quit the selected windows and icons.

PROPERTIES

Bring up Workspace Properties.

REFRESH

Refresh causes all windows on the screen to be repainted.

REREAD_MENU_FILE

Force an immediate rereading of the workspace menu customization file. **Olwm** will start a complete search for a menu file (as described in the *Menu Customization* section) and use the first valid file it finds.

RESTART

Restart the window manager by issuing an **exec(2)** on **argv**. This shouldn't affect any running applications, nor should it cause the server to shut down.

SAVE_WORKSPACE

Take a snapshot of the set of currently running applications, and put the command lines so obtained into the file ".openwin-init" in the user's home directory. This runs the command specified by the **SaveWorkspaceCmd** resource.

START_DSDM

Start providing the DSDM service. See the section on Drag and Drop for further information.

STOP_DSDM

Stop providing the DSDM service. See the section on Drag and Drop for further information.

WMEXIT

Exit the window manager without killing any applications.

Here is an example root menu specification.

"My Custom Menu" TITLE

```

Programs          MENU
  "Command Tool"  DEFAULT cmdtool
  "Text Editor"   textedit
  Mail            mailtool
  "File Manager"  filemgr
  Other          MENU
    "Other Tools" TITLE
    "Shell Tool"  shelltool
    "Icon Editor" iconedit
    Clock        clock
    "Perf Meter"  DEFAULT perfmeter
  Other          END
Programs          END PIN
"Repaint Screen" REFRESH
"Properties ..." PROPERTIES
Exit             EXIT

```

**COLORMAP
INSTALLATION**

Olwm will handle colormap installation for windows that have colormaps other than the default colormap. There are two colormap focus modes: "color-follows-mouse" and "color-locked". They are roughly analogous to the corresponding modes for input focus. However, the colormap focus mode can be completely independent of the input focus mode. The mode in which the system starts up is determined by the **ColorFocusLocked** resource.

Olwm keeps track of a set of windows that are eligible to have their colormaps installed. This set includes all top-level windows of clients. If any clients have specified other windows in a **WM_COLORMAP_WINDOWS** property, these windows are included in the set as well. The windows listed in this property need not be top-level windows; they can be nested subwindows as well.

In color-follows-mouse mode, **olwm** keeps track of the location of the pointer and always keeps installed the colormap of the eligible window underneath the pointer. Thus, you can install the colormap of a particular window simply by sliding the pointer into it. The default colormap will be restored if you move the pointer back out into a window frame or into the workspace. In this mode, the **WM_COLORMAP_WINDOWS** properties are tracked for changes, but only to change the set of eligible windows. Changes to these properties only cause colormaps to be installed if the eligible window under the pointer has changed as a result of the set of eligible windows changing. In this mode, no window is considered to have the colormap focus; colormap installation entirely is under control of the user.

In color-locked mode, colormaps are not installed based on pointer motion. Instead, colormaps are installed explicitly by the user using function keys or by a program changing the contents of the **WM_COLORMAP_WINDOWS** property on its top-level window.

The user can install the colormap of a window (or subwindow listed in the **WM_COLORMAP_WINDOWS** property) by moving the pointer over the window or subwindow and pressing the Color-Lock key (which is bound to Control-L2 by default). This will install the colormap of the window or subwindow under the pointer, and it will also grant the colormap focus to the top-level window. When a window has the colormap focus, **olwm** will honor changes to this window's **WM_COLORMAP_WINDOWS** property by installing the colormap of the first window named in this property. In this way, the application whose window has the colormap focus can control colormap installation by altering the contents of the **WM_COLORMAP_WINDOWS** property.

Note that, according to the ICCCM, if **WM_COLORMAP_WINDOWS** does not include the top-level window, it is assumed to occur first in the list. If you want your program to request colormap installation via changes to **WM_COLORMAP_WINDOWS**, you must make sure that the top-level window appears somewhere in this property. Otherwise, **olwm** will always install the colormap of the top-level window.

The colormap focus may be given to a window in several other ways. If you press the Color-Lock key over a window's title bar or border, that window will be given the colormap focus and the first window in the **WM_COLORMAP_WINDOWS** property will be installed. If the **AutoColorFocus** resource is set, new windows will be given the colormap focus automatically. If the **ColorTracksInputFocus** resource is set, the colormap

focus will always be given to the window that has the input focus.

If you press the Color-Lock key over the workspace, the default colormap will be installed, and any window with the colormap focus will lose it. The root window is then considered to have the colormap focus. At any time, you can revert to color-follows-mouse mode by pressing the Color-Unlock key. Any window with the colormap focus will lose it.

MAGNIFY HELP

Olwm provides on-line help through the Magnify Help utility for frames, icons, the Workspace and Window menus, window buttons, resize corners, pushpins, and the Workspace itself. This is done via a separate slave program, **olwmslave(1)**. The slave program is forked automatically when **olwm** starts up. The forking of the slave program can be controlled by the **RunSlaveProcess** resource.

MULTIPLE SCREENS

By default, **olwm** will manage windows on all screens of the display server. Most operations are unchanged from single screen operation. A window exists on a particular screen for its entire lifetime. The window cannot be moved from one screen to another, nor can it be resized to cross a screen boundary. Windows invoked from the Workspace menu will appear on the same screen as the menu. Magnify Help will appear on the same screen as the pointer when the Help key is pressed (F1 on x86 keyboards).

Previous releases required modifications to the user's **.xinitrc** script to start multiple instances of **olwm**, one for each screen. These modifications are no longer necessary. The default **Xinitrc** (which contains a single invocation of **olwm**) works for both single and multiple screen situations.

DRAG AND DROP

The OpenWindows drag and drop system relies on a third-party client (i.e. a client other than the source or destination clients of a drag and drop operation) to maintain a database of all possible locations on the screen where an object may be dropped. These locations are referred to as "drop sites." This third party client is thus called the Drop Site Database Manager or DSDM. By default, **olwm** is configured to provide the DSDM service to clients. This can be controlled using the **StartDSDM** resource or the **-dsdm** and **-nodsdm** command-line options.

If you have customized your Workspace Menu (see the section on Menu Customization) you can add items that use the **START_DSDM** and **STOP_DSDM** menu keywords. Invoking a menu item bound to one of these keywords will enable or disable **olwm**'s providing of the DSDM service.

A standalone client **dsdm(1)** exists in order to provide the DSDM service in the case where **olwm** is not running or if it has been directed not to provide the DSDM service. Note that the **START_DSDM** and **STOP_DSDM** functions do not run an actual **dsdm** process; rather, they control whether **olwm** provides the DSDM service itself. It is not necessary to run **dsdm** if **olwm** is providing the DSDM service.

GLOBAL RESOURCES

Global resources in **olwm** consist of two resource components. The first component in the resource name is taken from the trailing pathname component of **argv[0]**. This value is typically 'olwm'. This name can be altered by using the **-name** command-line argument. The second resource component names the global attribute being set. It should be

one of the names from the following list. Thus, to set the **AutoColorFocus** attribute, one would use "olwm.AutoColorFocus" as the resource specification.

Olwm will automatically pick up changes to many of these resources if the resource database changes at run-time. One can thus modify **olwm**'s behavior by changing the resource database with **xrdb(1)** or with Workspace Properties. If a resource value is specified on **olwm**'s command line, it will override the value in the resource database, and thus changing the resource's value in the database will have no effect on this resource setting.

Some resources are also interpreted by XView (see **xview(7)**) and are set by the Workspace Properties program (see **props(1)**). For these resources, **olwm** will also accept the string 'OpenWindows' as the first resource component. These resources are marked with an asterisk '*'.

Colors can be specified using the formats parsed by the Xlib XParseColor() function. Common formats are color names (see **showrgb(1)**) and explicit red, green, and blue values in hexadecimal, preceded by a '#'. For example, a cyan (full green and blue) would be specified with "#00ffff".

Boolean values can be specified with the words "true", "false", "on", "off", "yes", "no", "1", "0", "t", and "nil".

AutoColorFocus (*boolean*)

Indicates whether newly appearing windows are to be given the colormap focus automatically. See the section on Colormap Installation for further details. *Default value: false.*

AutoInputFocus (*boolean*)

Indicates whether newly appearing windows are to be given the input focus automatically. *Default value: false.*

AutoRaise (*boolean*)

Raise windows automatically when they receive the focus. This is useful in click-to-focus if you always like to type into the topmost window. This is useful in focus-follow-mouse when the **AutoRaiseDelay** resource is set to a reasonable value. *Default value: false.*

AutoRaiseDelay (*integer*)

Amount of time to delay, in microseconds, between a window receiving the focus and raising it above other windows. Effective only when the value of the **AutoRaise** resource is true. *Default value: 0.*

AutoReReadMenuFile (*boolean*)

Specifies whether the menu file is to be re-read whenever it changes. If the value is True, this will cause **olwm** to perform several **stat(2)** filesystem operations every time the Workspace Menu is raised. This may cause a noticeable delay. If this delay is objectionable, it may be eliminated by setting **AutoReReadMenuFile** to False. In such cases, one can use the **REREAD_MENU_FILE** to re-read the Workspace Menu definition file. See the section on Menu Customization for further information. *Default value: true.*

Background (*color*)

Specifies the background color. This is used for the background of masked icons. Note: it is not used for the backgrounds of icon windows such as those used by XView (see **xview(7)**). This resource is also distinct from the **WindowColor** resource. *Default value: white.*

BasicLocale (*locale name*)

Specifies the basic OPEN LOOK locale category setting. See the section on Locale Handling for more details.

Beep (*enumeration*) *

Specifies the circumstances under which **olwm** should beep. Permissible values are the strings "always", "never", and "notices". The string "never" means that **olwm** should never beep, "notices" means that **olwm** should beep only when a notice appears, and "always" means that **olwm** will beep whenever it is appropriate. *Default value: always.*

BorderColor (*color*)

Specifies the color used for window and icon borders. *Default value: black.*

ButtonFont (*font name*)

Font to be used for buttons in menus and notices. *Default value: lucidasans-12.*

ClickMoveThreshold (*integer*)

This value is used when bringing up a menu. If the pointer moves more than this amount while the menu button is down, the menu is considered to be in press-drag-release mode. Otherwise, the menu is in click-move-click mode. *Default value: 5.*

ColorFocusLocked (*boolean*)

Specifies the initial state of the colormap focus policy. If true, the default colormap is locked into the hardware. If false, the colormap of the window under the pointer is kept installed. See the section on Colormap Installation for further details. *Default value: false.*

ColorTracksInputFocus (*boolean*)

If true, indicates that the colormap focus is to be set automatically to any window that receives the input focus. See the section on Colormap Installation for further details. *Default value: false.*

CursorFont (*font name*)

Specifies the font to be used for cursors. It is probably not useful to change this unless you have an alternate cursor font with the same encoding as the OPEN LOOK cursor font. *Default value:*
`-sun-open look cursor-***-***-120-***-***-***.`

DefaultIconImage (*filename*)

Specifies a file containing a bitmap to be used as the default icon image.

DefaultIconMask (*filename*)

Specifies a file containing a bitmap to be used as the default icon mask.

DefaultTitle (*string*)

Specifies the string to be used in the title bar of windows that have not provided a string in the WM_NAME property. *Default value: No Name.*

DisplayLang (*locale name*)

Specifies the display language OPEN LOOK locale category. See the section on Locale Handling for more details.

DragRightDistance (*integer*) *

The number of pixels you must drag the mouse to the right in a menu item to bring up a sub-menu. The sub-menu always comes up when you move over the menu mark (the right-pointing triangle), regardless of the drag-right distance. *Default value: 100.*

DragThreshold (*integer*) *

This is the number of pixels the mouse must move while a mouse button is down in order to have the action be considered a drag. If the mouse moves fewer than this number of pixels while the button is down, it is considered to be click instead of a drag. *Default value: 5.*

DragWindow (*boolean*)

If true, drags the entire image of the window when you move it. Otherwise, just drags the window outline. *Default value: false.*

EdgeMoveThreshold (*integer*)

Specifies the amount of "hysteresis" provided when moving windows past the edge of the screen. When you move a window or an icon, it will pause when it touches the edge of the screen. This is to allow you to easily position windows right up against the edge of the screen. If you move farther, the window or icon will continue to move past the edge. You can prevent windows from ever lapping off the screen by setting an extremely large value (say, 10000) for this resource, and you can disable this feature entirely by specifying a value of zero. *Default value: 10.*

FlashCount (*integer*)

Number of times the title bar is flashed when the "Owners?" menu item is activated. *Default value: 6.*

FlashTime (*integer*)

Amount of time, in microseconds, for which the title bar is flashed when the "Owner?" menu item is activated. *Default value: 100000.*

FocusLenience (*boolean*)

If this is set to true, **olwm** will not enforce the ICCCM requirement that windows must have the input hint set in order to receive the input focus. This option is useful if you run clients that aren't ICCCM-compliant, like many X11R3-based clients. *Default value: false.*

Foreground (*color*)

Specifies the foreground color. This color is used mainly for the text of window and icon titles and in menus. *Default value: black.*

GlyphFont (*font name*)

Glyph font used for drawing OPEN LOOK graphics. Changing this font is mainly useful for changing its size. Specifying a different font, such as a text font, will result in undesirable behavior. *Default value:*
`-sun-open look glyph-*-*-*-*-*-*-*120-*-*-*-*-*-*-*`.

IconFlashCount (*integer*)

Number of times to flash the open/close "zoom" lines. *Default value:* 3.

IconFlashOffTime (*integer*)

Amount of time to pause, in microseconds, while open/close "zoom" lines are not visible. *Default value:* 1.

IconFlashOnTime (*integer*)

Amount of time to pause, in microseconds, while open/close "zoom" lines are visible. *Default value:* 20000.

IconFont (*font name*)

Font used for icon names. *Default:* lucidasans-12.

IconLocation (*enumeration*) *

One of the words "top-lr", "top-rl", "bottom-lr", "bottom-rl", "left-tb", "left-bt", "right-tb", or "right-bt". These specify that icons should be arranged along a particular edge of the screen, ordered from left to right or top to bottom as appropriate. The words "top", "bottom", "left", and "right" are synonyms for "top-lr", "bottom-lr", "left-tb", and "right-tb", respectively. *Default value:* bottom.

InvertFocusHighlighting (*boolean*)

In click-to-focus, the input focus is normally indicated by a solid rectangle in the title bar. In focus-follows-mouse, focus is normally indicated with two lines in the title bar. If this resource is true, the style of highlighting is inverted with respect to the focus style. This results in two lines for click-to-focus and a solid bar for focus-follows-mouse. *Default value:* false.

KeepTransientsAbove (*boolean*)

Specifies whether **olwm** should attempt to keep transient windows above their owner window. *Default value:* false.

KeyboardCommands (*enumeration*) *

Permissible values for this resource are **SunView1**, **Basic**, and **Full**. Values are case-sensitive. In **Full** mode, all OPEN LOOK Mouseless commands implemented by the window manager are active. See the section on Mouseless Navigation for further information. In **Basic** mode, the keys active are Open, Front, Help, and the colormap keys. In **SunView1** mode, the only keys active are Open and Front. *Default value:* Basic.

MenuAccelerators (*boolean*)

Determines whether menu accelerators are active. Used in conjunction with the **WindowMenuAccelerators** resource. Both must be set to true for menu accelerators to be active. *Default value:* true.

MinimalDecor (*list of strings*)

Specifies a list of windows that are to be decorated minimally. Decoration on

such windows includes only a thin border and resize corners, with no title bar or window button. The value should be a whitespace-separated list of strings. Each string should specify an application's class or instance name, as passed in the WM_CLASS property. Most applications set this property based on the name of the executable (i.e. argv[0]). For example, to specify that the clock and the calculator should be decorated minimally, you would use the following resource:

```
olwm.MinimalDecor: calctool clock
```

Many applications will allow you to override the value of the WM_CLASS property using the **-name** option on the command line. *Default value: (null).*

MouseChordMenu (*boolean*)

If true, uses a chorded mouse button combination for MENU instead of shift keys. See the Mouse Buttons section for further details. *Default value: false.*

MouseChordTimeout (*integer*)

Specifies the amount of time, in milliseconds, that **olwm** is to wait for subsequent events to disambiguate chorded mouse button event sequences. *Default value: 100.*

MultiClickTimeout (*integer*) *

The time, in tenths of a second, that differentiates a double-click from two single clicks. This value is also used to distinguish the click-move-click and press-drag-release modes of pop-up menus. If the MENU button is held down longer than this amount of time, the menu is considered to be in press-drag-release mode, otherwise it is considered to be in click-move-click mode. *Default value: 5.*

Numeric (*locale name*)

Specifies the numeric format OPEN LOOK locale category. See the section on Locale Handling for more details.

PaintWorkspace (*boolean*)

If true, **olwm** will use the **WorkspaceColor** resource to set the workspace (root window) background color. If false, **olwm** will not change the root window background. This is useful if you prefer to set your own workspace color using **xsetroot(1)** or a similar program. *Default value: true.*

PointerWorkspace (*boolean*)

If true, **olwm** will set the workspace (root window) cursor. If false, **olwm** will not change the root window cursor. This is useful if you prefer to set your own workspace cursor using **xsetroot(1)** or a similar program. *Default value: true.*

PPositionCompat (*boolean*)

Turns on backward compatibility for older applications that have a habit of always setting the PPosition flag in the WM_NORMAL_HINTS property, even when they haven't set a position. This most often occurs with X11R3-based clients. Without backward compatibility, these windows will always appear in the upper-left corner of the screen. With backward compatibility, these windows will be positioned according to the default OPEN LOOK window placement

policy, along the diagonal of the screen. This option will not affect windows that have a geometry specified on the command line. *Default value: false.*

PopupJumpCursor (*boolean*) *

Specifies whether to warp the cursor to pop-up windows. *Default value: true.*

PrintWarnings (*boolean*)

Determines whether **olwm** will issue non-fatal warning messages (such as X protocol errors) to its standard error file. *Default value: false.*

RaiseOnActivate (*boolean*)

Specifies whether a window is to be raised when it is activated via a Mouseless command. *Default value: true.*

RaiseOnMove (*boolean*)

Tells **olwm** to raise a window whenever it is moved by the user. *Default value: false.*

RaiseOnResize (*boolean*)

Tells **olwm** to raise a window whenever it is resized by the user. *Default value: false.*

RefreshRecursively (*boolean*)

Determines how the Refresh menu items on the window and workspace menus operate. If the value is true, **olwm** will walk the window hierarchy and send exposure events to every window. This is useful for refreshing windows that have backing store. If the value is false, **olwm** will map a window and then unmap it, causing all windows underneath that do not have backing store get exposures. When this feature is on, the Refresh operation generates a large amount of client-server traffic. It may be useful to turn this feature off if the connection transport has low bandwidth or high latency. *Default value: true.*

ReverseVideo (*boolean*)

If true, reverses the sense of black and white on monochrome screens. Ignored for color screens. *Default value: false.*

RubberBandThickness (*integer*)

Specifies the thickness of the "rubber-band" line that is drawn when a window is resized, when a group of windows is selected by dragging a rectangle on the root, and when a window is moved and the value of the **DragWindow** resource is false. *Default value: 2.*

RunSlaveProcess (*boolean*)

If false, disables the running of **olwmslave(1)** at startup time. If the slave process is not running, Magnify Help will not be available on objects owned by **olwm** such as pushpins and resize corners. *Default value: true.*

SaveWorkspaceCmd (*string*)

The command to execute to perform the Save Workspace functionality. This command defaults to running **owplaces(1)** which saves the currently running clients into the OpenWindows startup script `$HOME/.openwin-init`. *Default value:*

owplaces -silent -multi -local -script -tw -output \$HOME/.openwin-init

SaveWorkspaceTimeout (*integer*)

Number of seconds to wait while the Save Workspace operation is in progress. If the Save Workspace command has not completed within this amount of time, the operation is considered to have failed. *Default value: 30.*

SelectDisplaysMenu (*boolean*) *

If true, pressing the SELECT mouse button will bring up a menu item's sub-menu (if any) instead of executing the sub-menu's default action. *Default value: false.*

SelectionFuzz (*integer*)

Number of pixels of "fuzz" to be applied when selecting windows and icons by dragging a rectangle on the workspace. Consider an object that lies almost entirely within the selection rectangle, but that laps outside the rectangle by a few pixels. The object will be considered to be within the selection rectangle if it laps outside by fewer than or equal to "fuzz" pixels. *Default value: 1.*

SelectToggleStacking (*boolean*)

If true, double-clicking on a window will push it to the back instead of zooming it to and from its full size. *Default value: false.*

SelectWindows (*boolean*)

If false, the SELECT mouse button will not select windows and icons. Its other functions are unaffected. The ADJUST mouse button can still be used to select windows and icons. *Default value: true.*

ServerGrabs (*boolean*)

Controls whether **olwm** grabs the server while menus and notices are up. *Default value: true.*

SetInput (*enumeration*) *

Controls the input focus mode. If the value is "select", it means click-to-focus. If the value is "followmouse", it means focus-follows-mouse. *Default value: select.*

ShowMoveGeometry (*boolean*)

Indicates whether the geometry box should be shown while moving windows and icons. *Default value: false.*

ShowResizeGeometry (*boolean*)

Indicates whether the geometry box should be shown while resizing windows. *Default value: false.*

SnapToGrid (*boolean*)

Determines whether icons will snap to a grid when they are moved. *Default value: false.*

StartDSDM (*boolean*)

Determines whether **olwm** will provide the DSDM service. See the section on Drag and Drop for further details. *Default value: true.*

TextFont (*font name*)

Font used in the text of notices. *Default: lucidasans-12.*

TitleFont (*font name*)

Font used in title bars atop windows and menus. *Default: lucidasans-12 Bold.*

TransientsSaveUnder (*boolean*)

Specifies whether the save-under attribute of frames of transient windows is to be forced on. *Default value: false.*

TransientsTitled (*boolean*)

Specifies whether transient windows should have title bars. Normally, transient windows have a title bar and resize corners, but no window button or pushpin. Setting this resource to false will remove the title bar from transient windows. *Default value: true.*

Use3D (*boolean*)

Specifies whether to use 3D OPEN LOOK when possible. If false, 3D look is never used. If true, 3D is used unless the display hardware cannot support it. *Default value: true.*

Use3DFrames (*boolean*)

Specifies whether to use a 3D look for the frame borders. If true, the frames will be given a 3D look; otherwise, they have the same thick border as in 2D look. Some people prefer the look of 3D frames, but it is more difficult to distinguish selected from unselected windows with this option turned on. *Default value: false.*

Use3DResize (*boolean*)

Specifies whether the window resize corners are to be in the 3D look. If false, the 2D look is used for window resize corners. *Default value: true.*

WindowCacheSize (*integer*)

Olwm keeps a cache of windows in order to minimize unnecessary window creation and destruction. The value of this resource specifies the size of this cache. Setting this resource to zero disables the window cache. *Default value: 500.*

WindowColor (*color*) *

Specifies the color of windows. This is the "BG1" color for 3D OPEN LOOK. It is used for the backgrounds of windows, menus, and notices. The 3D effect is achieved by using highlight and shadow colors derived from this color. *Default value: #cccccc.* This specifies a 20% gray value.

WindowMenuAccelerators (*boolean*)

Determines whether menu accelerators are active. Used in conjunction with the **MenuAccelerators** resource. Both must be set to true for menu accelerators to be active. *Default value: true.*

WorkspaceBitmapBg (*color specification*)

Specifies the background color used for the workspace bitmap when the **WorkspaceStyle** resource is "tilebitmap". *Default value: black.*

WorkspaceBitmapFg (*color specification*)

Specifies the foreground color used for the workspace bitmap when the **WorkspaceStyle** resource is "tilebitmap". *Default value: white.*

WorkspaceBitmapFile (*filename*)

Specifies a X bitmap file that will be used for the workspace background when `WorkspaceStyle` is "tilebitmap". If the filename is not a full path name, the following directories are searched:

```
$OPENWINHOME/etc/workspace/patterns
$OPENWINHOME/include/X11/include/bitmaps
/usr/X11/include/X11/include/bitmaps
```

Default value: gray.

WorkspaceColor (*color*) *

Specifies the color for the workspace (root window). On startup, **olwm** will set the root window's background color to the color specified by this resource, and it will restore the default background on shutdown. To turn off this behavior, see the description of the **PaintWorkspace** resource. *Default value: #40a0c0*. This specifies a light blue color. *Note:* earlier versions of **olwm** would accept a bitmap file name as the value of the **WorkspaceColor** resource. This is no longer supported, and the **WorkspaceBitmapFile**, **WorkspaceBitmapBg**, and **WorkspaceBitmapFg** resources should be used instead.

WorkspaceStyle (*enumeration*)

This controls how the workspace is painted. If the value is "paintcolor", the solid color specified by the **WorkspaceColor** resource is used. If the value is "tilebitmap", the workspace is tiled with a bitmap using the **WorkspaceBitmapFile**, **WorkspaceBitmapBg**, and **WorkspaceBitmapFg** resources. If the value is "default", the server default root-weave pattern is used. If the value of the **PaintWorkspace** resource is false, then all of these resources are ignored and the workspace color or pattern is left unchanged. *Default value: paintcolor*

**SCREEN
RESOURCES**

In addition to the global resources described above, **olwm** also uses screen-specific resources. The first component of the resource specification is the trailing pathname component of `argv[0]`. The second component is the screen number appended to the string 'screen'. The screens are numbered sequentially starting from zero. The third component of the resource name is the name of the resource itself. For example,

```
olwm.screen1.ReverseVideo: true
```

enables reverse video on screen number 1 for **olwm**. To affect all screens, you can use resource wildcarding. For example, 'olwm*ReverseVideo: true' will set reverse video for all screens **olwm** manages.

Unlike many of the global resources, the screen-specific resources are only applied at **olwm** startup.

The following resources are available both globally and on a per-screen basis. A screen-specific resource overrides the corresponding global setting for that screen. Note that screen specific settings for `WorkspaceColor` and `WindowColor` will only affect **olwm**; this may cause clashes with XView clients which only use the global setting.

Background
 BorderColor
 Foreground
 ReverseVideo
 WindowColor
 WorkspaceColor

The following resources allow the selection of visuals other than the screen's default. Available visuals may be listed with the **xdpyinfo(1)** command.

Depth (*integer*)

Specify the visual depth to be used when searching for visuals. *Default value: none.*

Visual (*enumeration*)

Specify the visual class to be used when searching for visuals. Valid visual classes are **StaticGray**, **GrayScale**, **StaticColor**, **PseudoColor**, **TrueColor**, and **DirectColor**. Names are case-sensitive. *Default value: none.*

VisualID (*id*)

Specify the visual ID to be used. Note: specifying a visual by its ID is not portable, as IDs may vary from server to server and even from one invocation of a server to the next. *Default value: none.*

**MOUSELESS
 NAVIGATION**

Olwm implements OPEN LOOK Mouseless operations. This is a set of functions bound to keys that enable one to use the window system entirely without a pointing device. Some Mouseless functions are also useful for "cross-over" users, who may want to use them as accelerators for mouse-based operations. The full benefits of Mouseless operations are realized in click-to-focus mode, although the Mouseless operations can still be used in focus-follows-mouse mode.

To use the Mouseless functions, you must make sure that the **KeyboardCommands** resource value is "Full". Other settings for this resource will leave most of the Mouseless functions disabled. For further details, see the description of the **KeyboardCommands** resource in the Global Resources section. Enabling Mouseless operation only activates keyboard-based functions. It does not affect mouse functions in any way.

One can navigate from window to window using the Next Application, Previous Application, Next Window, and Previous Window functions, bound by default to Alt-n, Alt-Shift-n, Alt-w, and Alt-Shift-w, respectively. (See the section on Mouseless Navigation for more detailed information.) You can bring up both the window and the workspace menu using Alt-m and Alt-Shift-m, respectively. Once a menu is up, you can navigate through it by using the arrow keys or by pressing the first letter of the menu item you want to go to. You can execute the current item by pressing Return, or you can cancel the menu using Stop or Escape.

When Mouseless navigation is turned on, Move and Resize items will appear on the window menu. These items provide an alternative technique for moving and resizing windows. They can be invoked using the mouse, using the Mouseless menu navigation

functions from the keyboard, or by using Menu Accelerator keys (although they are not bound to any accelerator keys by default). After selecting either of these items, you will be put into a mode where you can move or resize the window using keyboard keys. In Move mode, you can use the arrow keys to move the window in the desired direction. You can also hold down the Control key to "jump" the window by a larger distance each time you press an arrow key. You can press Return to accept the new location, or you can press Escape or Stop to abort the move operation.

In Resize mode, the first arrow key selects the edge you are moving, and subsequent arrow keys move that edge. For example, to shrink a window from the right (that is, to move its right edge to the left) you would first enter resize mode, press the right arrow key to select the right edge, and then press the left arrow key to move this edge to the left. As in move mode, you can hold down Control to "jump" the edge by a greater increment. You can press Return to accept the new size, and you can press Escape or Stop to abort the resize operation.

MENU ACCELERATORS

Olwm supports accelerator keys for certain items on the Window Menu. By default, the items for which accelerators are enabled are Close (Meta-W) and Quit (Meta-Q). Pressing these key combinations will operate on the window or icon that has the input focus. Other Window Menu items are not bound to key combinations, but can be bound with resources. See the Key Binding Resources section (below) for further information. When a menu accelerator key is active for a particular function, an indication of this appears at the right edge of the menu item. Key combinations with modifiers are displayed in a self-evident fashion, except for the Meta modifier, which is displayed as a diamond mark. The meta keys are marked with diamonds on SPARC keyboards. On x86 keyboards the meta key is simulated by pressing the Control and Alt keys at the same time.

The default menu accelerator bindings may conflict with certain popular applications (such as Emacs or the Athena text widget). It is thus possible to disable menu accelerators on a per-application basis. To disable menu accelerators, add a resource of the form

```
olwm.Client.class.MenuAccelerators: false
```

to the resource database, where *class* is the application's class or instance name as written in the WM_CLASS property. For instance, to disable menu accelerators for Emacs, one would add the following

```
olwm.Client.Emacs.MenuAccelerators: false
```

to the **.Xdefaults** file.

KEY BINDING RESOURCES

Key bindings for mouseless navigation functions and menu accelerator keys are specified using resources. There is one resource per function, and the value of the resources are the keys to which the function is bound. The resource value consists of a comma-separated list of key specifications. Each key specification consists of a keysym optionally followed by modifier keysyms; the modifier keysyms are separated by '+' signs. For example, to bind a function to F2, control-F3, and alt-shift-F4, one would use the value:

F2,F3+Control,F4+Shift+Alt

Any keysym whose key is in the modifier mapping may be used as a modifier. The following can also be used as aliases for common modifier keysyms: **Shift, Lock, Control, Ctrl, Ctl, Meta, Alt, Super, and Hyper.**

Resource names are prefixed with the trailing pathname component of **argv[0]**, followed by **KeyboardCommand** for mouseless navigation functions, or **MenuAccelerator** for menu accelerator keys, followed by one of the resource names from the following list. (Note that the **KeyboardCommand** resource component is singular, and is not to be confused with the **KeyboardCommands** global resource name.) For example, the resource specification for setting the Stop function would be:

olwm.KeyboardCommand.Stop

and the resource specification for setting the Back menu accelerator function would be:

olwm.MenuAccelerator.Back

Each item in this list is followed by its default keyboard binding and a description of what the function does. Items marked with an asterisk '*' involve keyboard grabs. Items not marked with an asterisk are active only while **olwm** is in a mode, such as when a menu is up. Items marked with an exclamation point '!' are menu accelerators and are specified using the **MenuAccelerator** resource component as described above. Items not marked with an exclamation point are considered mouseless navigation functions and use the **KeyboardCommand** resource component.

Most of the mouseless navigation functions that use grabs are active only when the **KeyboardCommands** resource is set to **Full**. The menu accelerator functions all use grabs, and they are controlled by the global resources **MenuAccelerators** and **WindowMenuAccelerators**. For further information, see the description of these resources in the Global Resources section.

Stop (*L1, Escape*)

Abort the current mode or action.

DefaultAction (*Return, Meta-Return, Enter*)

Execute the default action for the current menu or notice.

Select (*space*)

Select the current button.

Adjust (*Alt-Insert*)

Toggle the selected state of the current object.

Menu (*Alt-space*)

Bring up a menu on the current object.

InputFocusHelp (*?, Control-?*)

Bring up Help on the object with the input focus.

| | |
|---|--|
| Up (<i>up-arrow</i>) | Move up one item. |
| Down (<i>down-arrow</i>) | Move down one item. |
| Left (<i>left-arrow</i>) | Move left one item. |
| Right (<i>right-arrow</i>) | Move right one item. |
| JumpUp (<i>Control up-arrow</i>) | Move up ten items. |
| JumpDown (<i>Control down-arrow</i>) | Move down ten items. |
| JumpLeft (<i>Control left-arrow</i>) | Move left ten items. |
| JumpRight (<i>Control right-arrow</i>) | Move right ten items. |
| RowStart (<i>Home, R7</i>) | Move to the start of the current row. |
| RowEnd (<i>End, R13</i>) | Move to the end of the current row. |
| DataStart (<i>Control-Home</i>) | Move to the start of the data. |
| DataEnd (<i>Control-End</i>) | Move to the end of the data. |
| FirstControl (<i>Control-I</i>) | Move to the first item. |
| LastControl (<i>Control-J</i>) | Move to the last item. |
| NextElement (<i>Tab, Control-Tab</i>) | Move to the next item. |
| PreviousElement (<i>Shift-Tab, Control-Shift-Tab</i>) | Move to the previous item. |
| Open (<i>Alt-L7</i>) * | Open the object with the input focus. |
| Help (<i>Help</i>) * | Bring up Magnify Help on the object under the pointer. |
| LockColormap (<i>Control-L2</i>) * | Install the colormap of the subwindow under the pointer, and give the colormap focus to the top-level window containing the pointer. See <i>Colormap Installation</i> for further details. |

- UnlockColormap** (*Control-LA*) *
Revert to color-follows-mouse mode, and unset colormap focus. See *Colormap Installation* for further details.
- Front** (*Alt-L5*) *
Bring the object with the input focus to the front.
- FocusToPointer** (*Alt-Shift-j*) *
Set the focus to the window under the pointer.
- NextApp** (*Alt-n*) *
Move the focus to the next base window. Windows are ordered clockwise starting at the top. Icons come after all windows, also in a clockwise fashion. Order proceeds from the last icon on a screen to the first window of the next screen. After the last screen, the order wraps back around to the first screen.
- PreviousApp** (*Alt-Shift-n*) *
Move the focus to the previous base window. See **NextApp** for details about the window traversal order.
- ToggleInput** (*Alt-t*) *
Move the input focus to the previous window that had the input focus.
- NextWindow** (*Alt-w*) *
Move to the next window in the family of windows consisting of a base window and a set of pop-up windows. Windows are ordered clockwise, starting at the top of the screen.
- PreviousWindow** (*Alt-Shift-w*) *
Move to the previous window in the family of windows consisting of a base window and a set of pop-up windows. Windows are ordered clockwise, starting at the top of the screen.
- TogglePin** (*Meta-Insert*) *
Toggle the state of the pin of the window with the input focus.
- SuspendMouseless** (*Alt-z*) *
Temporarily suspend all key grabs associated with Mouseless operation.
- ResumeMouseless** (*Alt-Shift-z*) *
Resume grabs after temporary suspension.
- QuoteNextKey** (*Alt-q*) *
Pass the next key sequence to the application with the focus, ignoring any grabs.
- Refresh** (*no binding*) *!
Repaint the window with the focus.
- Back** (*no binding*) *!
Move the focus window behind other windows.
- OpenClose** (*Meta-W*) *!
Toggle the open/close state of the window with the focus.
- FullRestore** (*no binding*) *!
Toggle the full-sized/normal-sized state of the window with the focus.

Quit (*Meta-Q*) *!

Quit the window with the focus.

Owner (*no binding*) *!

Flash the owner window of the pop-up window with the focus.

WorkspaceMenu (*Alt-Shift-m*) *

Bring up the workspace menu.

WindowMenu (*Alt-m*) *

Bring up the window menu on the window with the focus.

Move (*no binding*) *!

Move the window with the focus.

Resize (*no binding*) *!

Resize the window with the focus.

OpenClosePointer (*L7, Meta-w*) *

Toggle the open/close state of the window or icon under the pointer.

RaiseLower (*L5*) *

Raise the window under the pointer if obscured by other windows. Otherwise, lower the window if it obscures other windows.

MODIFIER CUSTOMIZATION

Olwm will alter the operation of certain mouse-based functions based on the state of the modifier keys. The relationship between the alteration and the associated modifier keys is controlled by a set of resources. Resource names are prefixed with the trailing path-name component of **argv[0]**, followed by **Modifier**, followed by a resource from the list below. For example, the resource specification to bind the Reduce modifier would typically be

```
olwm.Modifier.Reduce
```

The value of each resource is a comma-separated list of modifier keysyms. Each item in this list is followed by its default modifier and a description of what it does.

Constrain (*Control*)

Constrain a move or resize operation to be only on a horizontal or vertical direction.

Ignore (*Lock, NumLock, mod5, Mode_switch*)

The set of modifiers to be ignored when processing mouse button events. This resource should contain the set of locking modifiers, so that mouse actions are still interpreted properly even while locking modifiers are in effect. The *mod5* modifier is included in this set because XView places function keys into this row in the modifier mapping table for use with quick-move and quick-copy operations.

Invert (*Shift*)

When moving windows, temporarily invert the sense of the **DragWindow** resource. When resizing a window, temporarily move the window as long as

this modifier is held down. Return to resizing when the modifier is released.

Reduce (*Meta*)

When moving windows, reduce the amount of mouse motion by a factor of ten.

SetDefault (*Control*)

Sets the default item for a menu.

WMGrab (*Alt*)

Using the WMGrab modifier allows access to the mouse button functions anywhere over the window, not just over the window's title bar and border.

ENVIRONMENT

DISPLAY

Specifies the X11 server to which to connect.

LANG, LC_CTYPE, LC_MESSAGE, LC_TIME

These variables specify which locale to use when other methods of locale announcement are not available. (See the section on Locale Handling for more details.)

OLWMMENU

Specifies a file to use for the Workspace Menu.

OPENWINHOME

Specifies the location of the OpenWindows software.

FILES

\$HOME/.openwin-menu.localename

\$HOME/.openwin-menu

Contains the user-customized Workspace Menu specification.

\$OPENWINHOME/lib/openwin-menu.locale-name

\$OPENWINHOME/lib/openwin-menu

Contains the default Workspace Menu specification.

\$HOME/.openwin-init

Stores the command lines obtained during the Save Workspace operation.

\$OPENWINHOME/lib/app-defaults/Olwm

\$OPENWINHOME/lib/locale/locale-name/app-defaults/Olwm

Specifies system-wide default resource values.

TRADEMARKS

OPEN LOOK is a trademark of AT&T.

The X Window system is a trademark of the Massachusetts Institute of Technology.

OpenWindows is a trademark of Sun Microsystems, Inc.

REFERENCES

Rosenthal, David S.H. *Inter-Client Communication Conventions Manual for X11*. Copyright 1989 by the Massachusetts Institute of Technology. This document is commonly known as the ICCCM. It is an X Consortium Standard that specifies conventions to which all X11 clients must adhere.

OPEN LOOK Graphical User Interface Functional Specification. Copyright 1989 by Sun Microsystems, Inc. Addison-Wesley Publishing Company, Inc. ISBN 0-201-52365-5.
 OPEN LOOK Graphical User Interface International Extensions Functional Specification. Draft 1.1 (May 10, 1990). Copyright 1990 by Unix International.

SEE ALSO

dsdm(1), **olwmslave(1)**, **openwin(1)**, **owplaces(1)**, **props(1)**, **setlocale(3C)**, **xinit(1)**, **Xsun(1)**

NOTES

The resource names do not follow any classing structure. There is no general way to specify resources on a per-client basis.

There is no way to reconfigure the mouse buttons.

The uses of the modifier keys described in the Modifier Customization section interferes with the button bindings for one- and two-button mice. The default value of Modifier.Invert is Shift, which interferes with using shift-button1 for ADJUST. The default value of Modifier.Constrain is Control, which interferes with using control-button1 for MENU (on one-button mice only). One can set the Modifier.Invert and Modifier.Constrain resources to null (or to other modifiers) to avoid these conflicts, allowing full access to ADJUST and MENU on systems with one- and two-button mice. There is still a further conflict, as the default value of Modifier.SetDefault is also Control. Using control-button1 on a one-button system will bring up the menu, but will set the menu's default item. One must release the Control key after the menu is up in order to get normal menu operation. The choice of Alt as the default value for Modifier.WMGrab may conflict with some applications' key bindings.

The Exit menu item on the Workspace Menu doesn't really shut down the server. It kills off all clients being managed by the window manager, and then it exits the window manager itself. This works properly if some outside agent such as **xinit(1)** or **xdm(1)** is waiting for the window manager or a client to exit. The outside agent will take care of shutting down the server or reinitializing it. If you've started up the server a different way, this option may not work. Instead, the server will be left running with no clients and no window manager running, and you will have to login from elsewhere to kill the server. A common cause of this problem is an **.xinitrc** script that inadvertently leaves a non-windowed application (such as a daemon) running in the background. If the **.xinitrc** script ends with the **wait** shell command, it will never terminate. The fix is to change the script to either wait for a particular process-id, or to run the daemon in a subshell:

```
(daemon &)
```

Olwm is fairly simplistic about how it manages its keyboard bindings. For example, if you bind a function to control-F2, **olwm** will grab F2 with the Control modifier and with all combinations of the Lock and NumLock modifiers. If another locking modifier is in effect, **olwm**'s passive grab will not be activated, and thus the function will not work.

Olwm cannot manage multiple locales at one time, therefore all clients should be running in the same locale. The "C" locale is the exception. Applications using the "C" locale (such as non-internationalized applications) can be mixed with applications using one

other locale.

Olwm does not handle different sizes of the glyph fonts well. Each locale can define a different size for the default font (for example, the default glyph font size is 12 for the "C" locale and is 14 for the "japanese" locale). **Olwm** does not re-position the window decorations after switching locale, therefore the window decorations may appear to be wrong. To remedy this problem partially, **olwm** will not change the font when locale is switching from non-"C" locale to the "C" locale.

There is no input focus feedback for non-rectangular windows. The title string of non-rectangular windows cannot be displayed. Non-rectangular icon windows are not supported.

Olwm will not dynamically track screen-specific and client-specific resources. Changes to global resources, key binding resources, and modifier resources are applied dynamically.

The interaction of the **AutoColorFocus**, **ColorFocusLocked**, and **ColorTracksInputFocus** resources and the color locking and unlocking keys is overly complex.

Changing the Display Language locale setting or editing the menu specification file will cause **olwm** to unpin any menus that were pinned at the time.

Resources that specify time values use inconsistent units. Some resources are in tenths of a second, some are in milliseconds, and some are in microseconds.

| | |
|--------------------|--|
| NAME | olwmslave - helper program for olwm |
| SYNOPSIS | olwmslave |
| DESCRIPTION | olwmslave provides functionality for olwm (1) that is better done via a separate program, such as the Spot Help window. olwmslave is started by olwm and is not intended to be used directly. |
| SEE ALSO | olwm (1) |

| | |
|--------------------|---|
| NAME | openwin – OpenWindows startup script |
| SYNOPSIS | openwin [<i>options</i>] |
| DESCRIPTION | The openwin shell script sets up the proper environment for starting the entire OpenWindows package, including the Xsun (1) window server, olwm (1) an X11 ICCCM and OpenLook compliant window manager, and several default DeskSet(tm) tools. |
| OPTIONS | <p>There are several options which allow you to tailor the default setup of the Xsun server. openwin accepts the same command line options as Xsun. For more information, refer to both the Xsun(1) and Xserver(1) manual pages. Note that options passed to the server by the openwin shell script such as -r may be overwritten by settings specified by props(1) as it initializes the desktop. See the props(1) man page for more details.</p> <p>In addition, openwin accepts the following command line arguments that the server does not accept:</p> <p>[-server <i>Xsun</i>] This tells openwin which server binary to start. The default is \$OPENWINHOME/bin/Xsun.</p> <p>[-noauth] The OpenWindows server implements the "MIT-MAGIC-COOKIE" security mechanism which is a user-specific, rather than host-specific, authorization system. The default is to run with this authorization enabled. This option tells openwin to start the server without the "MAGIC COOKIE" authorization and will revert to no user-specific security. <i>Running the server with this option enabled lowers your level of security. It is recommended that this option not be enabled, except when explicitly needed.</i> See the OpenWindows documentation for more information.</p> <p>[-auth <i>protocol-name</i>] This option allows the user to choose the authentication protocol that the server will use to authenticate client connections. The two choices available are magic-cookie, or sun-des. The former sets the authentication protocol to "MIT-MAGIC-COOKIE" and is the default, while the latter makes the server use "SUN-DES-1", which is based on SecureRPC, as the authentication protocol. See the OpenWindows documentation for more information.</p> <p>[-includedemo] This option indicates that the path to the demo directory should be included in the user's search path (see PATH below).</p> <p>[-wm <i>wm-command</i>] This option allows the user to choose an alternate window manager to be started by the system default Xinitrc. The default is to start olwm(1) (the OpenLook window manager). For example, the command "openwin -wm twm" will start twm(1) instead of olwm. A <i>wm-command</i> of multiple words must be quoted. This option is likely to be ignored if the user has a \$HOME/.xinitrc script of their</p> |

own.

ENVIRONMENT

The behavior of **openwin** can also be tailored by means of the following environment variables:

OPENWINHOME

The **OPENWINHOME** variable is no longer required to be set. OpenWindows should always be accessible from **/usr/openwin**. If **OPENWINHOME** is set to anything other than **/usr/openwin**, OpenWindows will not start up properly.

DISPLAY

By default this is **":0"**, meaning this server is the first (zero based) one running on this machine. If you need to run more than one server on a given machine, use the **-display** option.

LD_LIBRARY_PATH

OpenWindows will no longer set **LD_LIBRARY_PATH**. Applications requiring the OpenWindows libraries should be compiled with **LD_RUN_PATH** set to **/usr/openwin/lib**. If you find an older binary which was compiled without **LD_RUN_PATH** set, you will need to set **LD_LIBRARY_PATH** to **\$OPENWINHOME/lib** in order to execute this program.

PATH

This is the path searched by shells (i.e. **csh(1)**, **sh(1)**, and **ksh(1)**) to find executable programs. **\$OPENWINHOME/bin** is prepended to your path if it is not there already.

MANPATH

openwin will add **\$OPENWINHOME/share/man** to **MANPATH** so that OpenWindows manual pages will be available to the user through the **man(1)** command.

HELPPATH

OpenWindows searches this path for help files, **\$OPENWINHOME/lib/help** is appended to any existing definition of **HELPPATH**.

OW_WINDOW_MANAGER

Specifies an alternate window manager command to be used by the system default Xinitrc script. This environment variable is private to the startup scripts and should not be used or set directly.

XINITRC

Specifies an init script executed by **xinit(1)** after the server is up and running. If **\$HOME/.xinitrc** does not exist and **XINITRC** is not already set, it is set to **\$OPENWINHOME/lib/Xinitrc**.

HOME

The name of the user's login directory, set by **login(1)** from the password file **/etc/passwd** (see **passwd(1)**).

FILES

\$HOME/.xinitrc

This is the file executed by **xinit** after the server is up and running. Previous releases copied **\$OPENWINHOME/lib/Xinitrc** into **\$HOME/.xinitrc**. This is no longer necessary and only users who want to customize server startup need to have this file.

\$HOME/.Xdefaults

This is where X11 application resources are defined. The default Xinitrc script will use **.Xdefaults** if it exists, or use **\$OPENWINHOME/lib/Xdefaults** if it does not exist.

\$OPENWINHOME/lib/Xinitrc

The default **xinit** init script. Site specific changes can be made here and will be used by users without an existing **.xinitrc**. The default Xinitrc contains the following:

```
# @(#)Xinitrc 1.20 92/12/08 OpenWindows startup script.

xrdb $OPENWINHOME/lib/Xdefaults # Load Default X11 resource
database
if [ -f $HOME/.Xdefaults ]; then
    xrdb -merge $HOME/.Xdefaults # Load Users X11 resource
database
fi

$OPENWINHOME/lib/openwin-sys # OpenWindows system
initialization

eval `locale_env -env` # Set Locale Environment

if [ "$OW_WINDOW_MANAGER" ]; then # Alternate Window Manager
    xsetroot -def # Clear root window
    $OW_WINDOW_MANAGER & wmpid=$! # Start Alt Window Manager
    dsdm & # OpenLook Drop Site Database
    unset OW_WINDOW_MANAGER
else
    sleep 15 & pid=$! # OpenLook Window Manager
    olwm -syncpid $pid & wmpid=$!
    wait $pid # Pause until olwm inits
fi

if [ -x $HOME/.openwin-init ]; then
    $HOME/.openwin-init # Custom OpenWindows tools
else
    $OPENWINHOME/lib/openwin-init # Default OpenWindows tools
fi
```

wait \$wmpid

Wait for wm (key client) to exit

\$OPENWINHOME/lib/Xdefaults

Site specific changes to the default resource database can be made here and will be propagated to users without an existing **.Xdefaults**.

\$HOME/.Xauthority and **\$HOME/.xsun.hostname:displaynumber**

Files where per-session authorization information is written.

SEE ALSO

Xsun(1), Xserver(1), login(1), olwm(1), passwd(1), props(1), xinit(1), xrdb(1)

| | |
|--------------------------------------|--|
| NAME | owplaces – list client applications running on a display |
| SYNOPSIS | <pre>owplaces [-display <i>displaystring</i>] [-timeout <i>nseconds</i>] [-single -multi -pointer] [-all -local -remote -host <i>hostname</i>] [-script] [-output <i>filename</i>] [-ampersand] [-tw] [-silent]</pre> |
| DESCRIPTION | owplaces is a tool to list the command lines for the selected clients. The output is geared towards use in a startup script such as \$HOME/.openwin-init |
| OPTIONS | <p>The basic options are as follows:</p> <p>-display <i>string</i> Specify the name of the display to manage. Overrides the DISPLAY environment variable, if any.</p> <p>-timeout <i>nsecs</i> Specifies the time to wait for a reply for those clients that are updating their command lines. Default is 30 seconds.</p> |
| MODE OPTIONS | <p>Options that control which clients are printed:</p> <p>-single Search only the default screen of the specified display for clients.</p> <p>-multi Search all screens of the specified display for clients. This is the default.</p> <p>-pointer Print only the client selected by clicking the mouse in the desired window.</p> |
| CLIENT MACHINE FILTER OPTIONS | <p>Options to filter clients based on host machine (as specified by the WM_CLIENT_MACHINE property):</p> <p>-all Print clients on all hosts. This is the default.</p> <p>-local Print clients local to this machine only.</p> <p>-remote Print clients that are remote to this machine.</p> <p>-host <i>hostname</i> Print clients from <i>hostname</i> only.</p> |
| INITIALIZATION SCRIPT OPTIONS | <p>Options useful to producing shell scripts:</p> <p>-script Instead of printing command lines, output will be a Bourne shell script suitable for use as an initialization file. The script can handle multiple screens if -multi is also specified.</p> <p>-output <i>filename</i> Directs output to <i>filename</i> instead of stdout. If <i>filename</i> exists, it is renamed as <i>filename.BAK</i>.</p> <p>-ampersand Appends an ampersand character (&) to end of each command line. This is implied if -script is used.</p> <p>-tw Prepends the command toolwait(1) to each client. The -ampersand option has no affect if this option is specified.</p> |

-silent Suppress error messages.

EXAMPLES To create a **\$HOME/.openwin-init** startup script just as **olwm(1)** does for "Save Workspace":

```
example% owplaces -multi -script -output $HOME/.openwin-init
```

SEE ALSO **olwm(1)**, **xlsclients(1)**, **toolwait(1)**

| | |
|--------------------|--|
| NAME | pageview – PostScript language previewer for OpenWindows |
| SYNOPSIS | <pre> pageview [-mono] [-w <i>paperwidth</i>] [-h <i>paperheight</i>] [-dpi <i>dots/inch</i>] [-mcd <i>colordensity</i>] [-page <i>pagenumber</i>] [-dir <i>directory</i>] [-left right upsidedown] [-timeout <i>job_timeout (sec)</i>] [-aa] [-low_memory] [-usage] [-v] [-verbose] [<i>psfile</i> -] </pre> |
| DESCRIPTION | <p>pageview is an interactive POSTSCRIPT previewer. pageview renders a document, a page at a time, onto an offscreen bitmap of arbitrary size, resolution and orientation. You can then adjust the viewing window's size to see as much of the page as desired. The mouse buttons are used to position the page under the window in two different modes.</p> <p>The left button moves the page in "relative mode". This allows you to move the page in a physically intuitive way. You press the left button on the page and while you drag the mouse around, the spot on the page that was under the mouse cursor when you pressed the button remains stationary relative to the cursor.</p> <p>The middle button moves the page in "absolute mode". This allows you to easily get to the edges of the document, especially when the DPI is large and/or the window is small. When you press the middle button on a point in the window, the page is adjusted so that the same relative point on the page is under the mouse. For example, if you press the middle button at the top right corner of the window, you will see the top right corner of the page. A little experimentation with a page at 300 dpi and you will find this mode indispensable.</p> <p>The three menu buttons across the top of the main window are described below:</p> <p>File The File menu is used to bring up the Load... or Print... dialogs.</p> <p>Load... brings up a dialog which prompts for a directory and filename to load a new POSTSCRIPT file.</p> <p>Print... brings up a dialog which prompts for the name of the printer to send the POSTSCRIPT document to. You can print the whole document or only the current page. You may also write the POSTSCRIPT to a named file.</p> <p>View The View menu allows you to move to the Next, Previous, First and Last pages of multipage documents.</p> <p>Edit The edit menu has two choices, POSTSCRIPT and Properties.</p> <p>PostScript... brings up a text editor with the POSTSCRIPT document in it and a window which contains all of the errors and other output from the document. You may make changes to the document and press the run button to re-render the page.</p> <p>Properties... brings up the property sheet for page size, orientation and resolution.</p> <p>DPI: This lets you change the "resolution" of the retained bitmap which the page is being rendered onto. 72 dpi will make a US Letter sized page be 612x792 pixels, where 300 dpi would be 2550x3300 pixels. This has the effect of making 72</p> |

dpi images appear smaller and 300 dpi pages appear larger due to the static resolution of the display. **pageview** starts out at 85 dpi, unless you have the environment variable **\$DPI** set to some other default, or you use the **-dpi** command line argument.

Size: This lets you change the size of the retained bitmap which the page is rendered onto. USLetter is 8.5x11", Legal is 8.5x14", and the European page sizes are, A3 29.7 x 42.0 cm, A4 21.0 X 29.7 cm, A5 14.8 x 21.0 cm, and B5 17.6 x 25.0 cm. These values can be set to custom values by the **-height** and **-width** command line arguments.

Orientation:

This menu lets you choose which way to rotate the paper in 90 degree increments. This is useful for viewing slides which are commonly rendered in "Landscape left" orientation.

Timeout: This option allows you to select the job timeout value. This value is used by **pageview** as the amount of time in which a page of a document must be rendered onto the screen. If the page cannot be rendered in this time, **pageview** assumes that there is some sort of POSTSCRIPT error on the page which is causing this problem (such as a string or procedure that does not end) and stops trying to display the page. However, often times, documents that contain complex graphics may take a longer than average amount of time to display. In this case, you should increase the job timeout value and attempt to display the page again.

Ignore PostScript Structuring Comments:

pageview is designed to scan a POSTSCRIPT file and determine the different sections of it based on POSTSCRIPT Structuring Comments. For example, from reading these comments, **pageview** determines where each page of the document starts and ends. However, as there are many POSTSCRIPT documents that do not use the comments, **pageview** often gets confused as to the number of pages in a document and/or where they began and end. Therefore, if you attempt to view a document, but feel that the resulting displayed document does not look as you expected, you may want to turn on this option. This option will assume that there are no POSTSCRIPT Structuring comments, and interpret the file differently. Be aware though, this method of interpreting the POSTSCRIPT file may be slower than the default method, especially if the file is very big.

OPTIONS

-mono

is used to force **pageview** to use a monochrome retained canvas on color systems. This saves memory and is faster on some framebuffers.

-w *paperwidth*

sets the width of the "paper" to *paperwidth* inches, the default is 8.5.

-h *paperheight*

sets the height of the "paper" to *paperheight* inches, the default is 11.

- dpi** *dots/inch*
sets the "dpi" of the "paper" to *dots/inch*. The environment variable **\$DPI** is used if this option is not present, and the default is 85 if this variable is not in the environment. Caution must be used in setting this argument as well as the paper size args above, so you do not exhaust memory resources. For example a USLetter sized page previewed at 300 dpi, takes up 300*8.5/8*300*11 or a little over a Megabyte. The same page at 1500 dpi takes over 26 Megabytes.
 - mcd** *colordensity*
Sets the maximum color density. The default maximum color density is 100. However, if an image is to be viewed with a DPI greater than 100, it will not be displayed in color unless the maximum color density is set greater than 100.
 - page** *pagenumber*
Sets the page number of the document that is to be displayed when **pageview** starts.
 - dir** *directory*
Sets the current working directory to *directory* so that you can type filenames at the "Load File: " prompt relative to *directory*.
 - left | right | upsidedown**
Sets the rotation of the page.
 - timeout** *job_timeout (sec)*
Sets the job timeout value. The default job timeout value is 30 seconds.
 - aa** Turns on antialiasing.
 - low_memory**
Runs a low memory version of **pageview**. This version does not create the offscreen bitmap, and rather renders the document directly onto the viewing canvas. This uses less memory, but performance may suffer especially when the **pageview** window is hidden and then exposed, as the entire page must be rendered again. If the user attempts to move the page in "relative mode" (pan), the offscreen bitmap is automatically created as this function is not possible without it. Also, this option cannot be used if antialiasing is turned on.
 - usage** Prints out all valid command line options.
 - v** Prints out the current version of **pageview**.
 - verbose** Prints lots of debugging information (not useful to the user)
- If **psfile** is specified, the POSTSCRIPT code is taken from that file.
If no argument is given, **pageview** comes up with no document in it and if a '-' is given as the argument, **pageview** reads the POSTSCRIPT program from standard input.

SEE ALSO

lp(1)
OpenWindows user documentation
POSTSCRIPT Language Reference Manual, Adobe Systems Inc., Addison-Wesley

TRADEMARK

POSTSCRIPT is a registered trademark of Adobe Systems Inc

| | |
|---------------------|---|
| NAME | perfmeter – display system performance values in multiple dials or strip charts |
| SYNOPSIS | perfmeter [-a] [-d] [-g] [-h <i>h-hand-int</i>] [-l] [-m <i>m-hand-int</i>] [-n <i>sample-filename</i>] [-name <i>app-name</i>] [-p <i>page-length</i>] [-s <i>sample-time</i>] [-t <i>value</i>] [-v] [-C <i>chart ceiling</i>] [-H] [-L] [-M <i>chart smax minmax maxmax</i>] [-S] [-V] [-Wn] [+Wn] [<i>hostname</i>] |
| AVAILABILITY | This command is available with the OpenWindows environment. |
| DESCRIPTION | <p>perfmeter is an OpenWindows XView utility that displays performance values (statistics) for a given <i>hostname</i>. If no host is specified, statistics on the current host are metered. The rstatd(1M) daemon must be running on the machine being metered.</p> <p>perfmeter has two display formats. You can display performance values in the form of strip charts (the default). Alternatively, the performance information can be displayed in the form of multiple meter dials. By default, the display is updated with a <i>sample-time</i> of 2 seconds. The hour hand of a meter represents the average over a 20-second interval; the minute hand, the average over 2 seconds. The default value displayed is the percent of CPU being utilized.</p> <p>The maximum scale value for the strip chart will automatically double or halve to accommodate increasing or decreasing values for the host machine. This scale can be restricted to a certain range with the -M option.</p> |
| OPTIONS | <p>-a Display all the performance values simultaneously.</p> <p>-d Start the perfmeter display, by showing metered dials.</p> <p>-g Start the perfmeter display, by showing strip charts.</p> <p>-h <i>h-hand-int</i> Set the hour-hand interval to <i>h-hand-int</i> seconds.</p> <p>-l Log the perfmeter samples to a file.</p> <p>-m <i>m-hand-int</i> Set the minute hand interval to <i>m-hand-int</i> seconds.</p> <p>-n <i>sample-filename</i> Name of the file to use, when saving perfmeter samples.</p> <p>-name <i>app-name</i> This option specifies the application name under which resources are to be obtained, rather than the default executable file name. <i>app-name</i> should not contain “.” or “*” characters.</p> <p>-p <i>page-length</i> Sets the page length for perfmeter samples written to file. The default is 60 lines per page.</p> <p>-s <i>sample-time</i> Set the sample time to <i>sample-time</i> seconds.</p> |

-t value

This option can be present multiple times, and it determines which performance value are to be monitored. The following options are available:

- cpu** Percent of CPU being utilized.
- pkts** Ethernet packets per second.
- page** Paging activity in pages per second.
- swap** Jobs swapped per second.
- intr** Number of device interrupts per second.
- disk** Disk traffic in transfers per second.
- cntxt** Number of context switches per second.
- load** Average number of runnable processes over the last minute.
- colls** Collisions per second detected on the ethernet.
- errs** Errors per second on receiving packets.

-v Show version number

-C chart ceiling

Set a ceiling value for a given strip chart. If the value goes beyond this ceiling and **perfmeter** is running on a color screen, then this portion of the strip chart will be displayed in red.

-H Multiple dials or strip charts will be displayed initially in a horizontal direction.

-L If the **-g** option is present, then initially display the strip charts in line format direction.

-M chart smax minmax maxmax

Set a range of maximum values for the given strip chart. Values for each of the arguments should be powers of 2. *smax* sets the starting maximum-value. *minmax* sets the lowest allowed maximum-value for the scale. *maxmax* sets the highest allowed maximum-value.

-S If the **-g** option is present, then initially display the strip charts in a solid filled format.

-V Multiple dials or strip charts will be displayed initially in a vertical direction.

-Wn Start **perfmeter** with no title line.

+Wn Start **perfmeter** with a title line present.

USAGE Commands

You can add extra values to be monitored by clicking the MENU mouse button, and selecting from the popup menu. Alternatively, you can select **Properties** from this menu, and this will bring up the property sheet for the tool. The property sheet will allow you to modify what the **perfmeter** is monitoring, which direction graphs or dials are displayed in, whether to display dials or graphs, if the graph is solid or lined, whether to monitor a local machine or a remote one, and how often it checks.

Note that **perfmeter** does not try to do a best fit, given the shape of the window and the number of graphs you wish to display. For example, if the display direction is vertical, and you are currently displaying ten graphs in a tall narrow window, then resizing the window to be short and wide will result in ten very thin wide graphs.

Most meter parameters can be modified through the use of keyboard accelerators. Move the pointer into the window and type one of the following keyboard keys:

- d** Toggle the direction the graphs or dials are displayed in, between horizontal and vertical.
- g** Toggle the display between graphs and dials.
- h** Decrease *hourhandintv* by one
- H** Increase *hourhandintv* by one
- m** Decrease *minutehandintv* by one
- M** Increase *minutehandintv* by one
- q** Quit the **perfmeter**
- s** If graphs are being displayed, toggle between lined and solid graphs.
- S** Toggle the saving of samples to file.
- t** Toggle between the monitoring of the local machine and a remote machine. Note that a remote machine name must have already been setup via the tools property sheet.
- 1-9** Set *sampletime* to a range from 1 to 9 seconds.

On startup, **perfmeter** will use the following X resources:

- Resource:** deskset.perfmeter.displayGraph
- Values:** True, False (True)
- Description:** Indicates whether **perfmeter** should be started showing strip graphs or metered dials.

- Resource:** deskset.perfmeter.hourInterval
- Values:** Hour hand interval (numeric)
- Description:** Set the hour-hand interval to this number of seconds.

- Resource:** deskset.perfmeter.minInterval
- Values:** Minute hand interval (numeric)
- Description:** Set the minute-hand interval to this number of seconds.

- Resource:** deskset.perfmeter.sampleTime
- Values:** Sample time (numeric)
- Description:** The interval in seconds, between which samples are taken.

- Resource:** deskset.perfmeter.resizeGraphView
- Values:** True, False (True)
- Description:** Indicates whether changing the number of graphs being monitored,

| | |
|---------------------|--|
| | should cause the window to be resized. |
| Resource: | deskset.perfmeter.showLocal |
| Values: | True, False (True) |
| Description: | Indicates whether the local or the remote host should be initially displayed. |
| Resource: | deskset.perfmeter.monitor |
| Values: | Values to monitor (string) |
| Description: | A comma separated list of different values to monitor. These values are identical to their command line equivalents. |
| Resource: | deskset.perfmeter.machine |
| Values: | Machine to monitor (string) |
| Description: | The machine for which the strip graphs or metered dials will be displayed. |
| Resource: | deskset.perfmeter.displayVertical |
| Values: | True, False (True) |
| Description: | Indicates whether the strip graphs or metered dials will be displayed in a horizontal or vertical direction. |
| Resource: | deskset.perfmeter.graphSolid |
| Values: | True, False (False) |
| Description: | If the strip graphs are being displayed, this resource indicates whether lined or solid graphs are displayed. |
| Resource: | deskset.perfmeter.cpuMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the cpu graph or dial. |
| Resource: | deskset.perfmeter.pktsMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the pkts graph or dial. |
| Resource: | deskset.perfmeter.pageMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the page graph or dial. |
| Resource: | deskset.perfmeter.swapMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the swap graph or |

| | |
|---------------------|---|
| | dial. |
| Resource: | deskset.perfmeter.intrMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the intr graph or dial. |
| Resource: | deskset.perfmeter.diskMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the disk graph or dial. |
| Resource: | deskset.perfmeter.cntxtMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the cntxt graph or dial. |
| Resource: | deskset.perfmeter.loadMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the load graph or dial. |
| Resource: | deskset.perfmeter.collMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the coll graph or dial. |
| Resource: | deskset.perfmeter.errMaxValues |
| Values: | Starting, lowest and highest maximum values (numeric) |
| Description: | The starting, lowest and highest maximum values for the err graph or dial. |
| Resource: | deskset.perfmeter.collectWhenObscured |
| Values: | True, False (True) |
| Description: | Indicates whether data will be monitored and cached, when the perfmeter window is fully obscured. |
| Resource: | deskset.perfmeter.hasTitle |
| Values: | True, False (False) |
| Description: | Indicates whether the perfmeter window has a title line. |
| Resource: | deskset.perfmeter.labelFont |
| Values: | Font name string |
| Description: | The name of the font used to display all the labels for the dials and graphs. |

| | |
|---------------------|--|
| Resource: | deskset.perfmeter.saveSamples |
| Values: | True, False (False) |
| Description: | Indicates whether perfmeter should save monitored sample values to disk. |
| Resource: | deskset.perfmeter.sampleFilename |
| Values: | Filename (string) |
| Description: | Name of the file to save sample in. If no filename is given, perfmeter creates a unique filename in the users home directory. |
| Resource: | deskset.perfmeter.samplePageLength |
| Values: | Sample file page length (numeric) |
| Description: | The page length (in lines) in the sample file. If no value is given, the default is 60. |
| Resource: | deskset.perfmeter.cpuColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the cpu strip chart. |
| Resource: | deskset.perfmeter.pktsColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the pkts strip chart. |
| Resource: | deskset.perfmeter.pageColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the page strip chart. |
| Resource: | deskset.perfmeter.swapColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the swap strip chart. |
| Resource: | deskset.perfmeter.intrColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the intr strip chart. |
| Resource: | deskset.perfmeter.diskColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the disk strip chart. |
| Resource: | deskset.perfmeter.cntxtColor |
| Values: | Color name string or hexadecimal color specification string |
| Description: | The color of the cntxt strip chart. |
| Resource: | deskset.perfmeter.loadColor |
| Values: | Color name string or hexadecimal color specification string |

| | |
|--------------------|--|
| Description | The color of the load strip chart. |
| Resource: | deskset.perfmeter.collColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the coll strip chart. |
| Resource: | deskset.perfmeter.errColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color of the err strip chart. |
| Resource: | deskset.perfmeter.ceilingColor |
| Values: | Color name string or hexadecimal color specification string |
| Description | The color used when any strip chart goes above its ceiling value. |
| Resource: | deskset.perfmeter.ceilingStyleSolid |
| Values: | True, False (False) |
| Description | Indicates how the graph line should be drawn on solid graphs when the current value is above the ceiling value. Setting this resource true indicates that the full line will be drawn in the ceiling color; setting it false means that only the portion above the ceiling value will be drawn in the ceiling color. |
| Resource: | deskset.perfmeter.cpuCeiling |
| Values: | cpu ceiling value (numeric) |
| Description | The ceiling value to use when displaying the cpu strip chart. Values above this will be displayed in a different color on color screens. |
| Resource: | deskset.perfmeter.pktsCeiling |
| Values: | pkts ceiling value (numeric) |
| Description | The ceiling value to use when displaying the pkts strip chart. Values above this will be displayed in a different color on color screens. |
| Resource: | deskset.perfmeter.pageCeiling |
| Values: | page ceiling value (numeric) |
| Description | The ceiling value to use when displaying the page strip chart. Values above this will be displayed in a different color on color screens. |
| Resource: | deskset.perfmeter.swapCeiling |
| Values: | swap ceiling value (numeric) |
| Description | The ceiling value to use when displaying the swap strip chart. Values above this will be displayed in a different color on color screens. |
| Resource: | deskset.perfmeter.intrCeiling |
| Values: | intr ceiling value (numeric) |
| Description | The ceiling value to use when displaying the intr strip chart. Values |

above this will be displayed in a different color on color screens.

Resource: deskset.perfmeter.diskCeiling
Values: disk ceiling value (numeric)
Description: The ceiling value to use when displaying the disk strip chart. Values above this will be displayed in a different color on color screens.

Resource: deskset.perfmeter.cntxtCeiling
Values: cntxt ceiling value (numeric)
Description: The ceiling value to use when displaying the cntxt strip chart. Values above this will be displayed in a different color on color screens.

Resource: deskset.perfmeter.loadCeiling
Values: load ceiling value (numeric)
Description: The ceiling value to use when displaying the load strip chart. Values above this will be displayed in a different color on color screens.

Resource: deskset.perfmeter.collCeiling
Values: coll ceiling value (numeric)
Description: The ceiling value to use when displaying the coll strip chart. Values above this will be displayed in a different color on color screens.

Resource: deskset.perfmeter.errCeiling
Values: err ceiling value (numeric)
Description: The ceiling value to use when displaying the err strip chart. Values above this will be displayed in a different color on color screens.

FILES /etc/inetd.conf starts statistics server

SEE ALSO **xview(7)**, **netstat(1M)**, **rstatd(1M)**, **vmstat(1M)**
Solaris User's Guide
 "About Performance Meter" in the Help Handbook available through the Help option on the Workspace menu.

| | |
|--------------------|---|
| NAME | plaid – paint some plaid-like patterns in an X window |
| SYNOPSIS | plaid [-b] [-fg <i>color</i>] [-bg <i>color</i>] [-bd <i>color</i>] [-bw <i>number</i>] [-geometry <i>geometry</i>] [-display <i>display</i>] |
| OPTIONS | -b enable backing store for the window -fg <i>color</i> This option specifies the color to use for the foreground of the window. The default is “white.” -bg <i>color</i> This option specifies the color to use for the background of the window. The default is “black.” -bd <i>color</i> This option specifies the color to use for the border of the window. The default is “white.” -bw <i>number</i> This option specifies the width in pixels of the border surrounding the window. -geometry <i>geometry</i> define the initial window geometry; see X11(7) . -display <i>display</i> specify the display to use; see X11(7) . |
| DESCRIPTION | Plaid displays a continually changing plaid-like pattern in a window. |
| SEE ALSO | X11(7) |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |

| | |
|--------------------|---|
| NAME | printtool – OpenWindows tool for printing files |
| SYNOPSIS | printtool [-P printer] [-v] [-?] |
| DESCRIPTION | <p>Printtool is a DeskSet utility that provides a Graphical User Interface (GUI) for the lp(1), cancel(1), and lpstat(1) printing functions.</p> <p>Printtool is an XView-based OpenWindows application that allows users to easily control standard printing options such as the name of the file to be printed, the name of the printer to use, whether or not to print a header page, number of copies to print, print method, etc. Printtool also lets users determine the status of print jobs in progress and cancel print jobs that are waiting in the queue.</p> |
| OPTIONS | <p>In addition to the generic tool arguments supported by xview(7), printtool accepts the following options:</p> <p>-P printer Send output to the named printer.</p> <p>-v Show the version number and the usage message of this release of the printtool program.</p> <p>-? Same as -v</p> |
| USAGE | <p>Printtool is part of the OpenWindows user environment. For more information on the basic functions of OpenWindows, see the <i>Solaris User's Guide</i></p> <p>Users will normally use printtool to print files by entering the name of a file in the Filename field, selecting a printer from the scrollable list of possible printers and then clicking on the Print button. The number of copies to print is set to 1 by default but can be easily changed by using the up/down arrows in the Copies control or by entering a number. The Header check-box allows a user to specify whether or not a header page should be printed as part of the print job. The setting of the Header check-box will override the setting in the Properties... window.</p> <p>Users may also drag and drop file icons from filemgr(1) or mail attachments from mail-tool(1) onto printtool's drop target instead of typing a name in the Filename: text field. The drop target is a small rectangular area in the upper right-hand corner of the printtool window. If the printtool window is not open users can simply drag and drop a file icon onto the icon for printtool itself.</p> <p>The status of the printer queue can be checked automatically or by clicking the Status button. Print jobs may be terminated by selecting their name from the Job scrollable list and then clicking on the Stop Printing button.</p> <p>A print method can be invoked interactively by using the "Override Default Print Method" check-box in the Properties... window and entering the print method in the space provided. This option allows finer control over how files are to be printed. Printtool will not add or append anything to or from the print method. Users can use the environment variables \$FILES, \$PRINTER, \$LPDEST and \$COPIES, which are specified in printtool, to write their own print method. For instance, the entry:</p> |

mp -l

will pass the file through the **mp(1)** filter with the "landscape" option selected. This is the equivalent of executing the following command on the command line:

```
cat $FILE | mp -l | lp
```

The **Information...** button brings up a text window with information about the selected printer.

The **Properties...** button brings up a window that allows users to specify global defaults for **printtool**. These include whether or not to print header pages, whether to beep or flash when all jobs done, and whether or not to override default print method. The **Apply** button only sets these options for the current process. To save the current settings permanently the **Save as Defaults** button should be clicked.

RESOURCES

On startup, **printtool** uses the following resources which are stored in **\$HOME/.desksetdefaults**:

| | |
|--------------------|---|
| Resource: | deskset.printtool.checkInterval |
| Values: | Status check interval (numeric) |
| Description | This is the number of seconds between status checks on the printer. The default value is 10 seconds. |
| Resource: | deskset.printtool.headerPage |
| Values: | true, false |
| Description | This is the flag for printing with header page or not. The default value is false (no header page). |
| Resource: | deskset.printtool.printMethodOverride |
| Values: | true, false |
| Description | This is the flag for overriding the default print method. The default value is false (use default print method which is determined by Classing Engine). |
| Resource: | deskset.printtool.lastPrintMethod |
| Values: | Print method string or full path of script string |
| Description | This is the print method for printing instead of the default method if deskset.printtool.printMethodOverride is set to true. |
| Resource: | deskset.printtool.notifyFlash |
| Values: | true, false |
| Description | If set to true, printtool will flash the window or icon when all print jobs are done. The default value is false (no flash). |
| Resource: | deskset.printtool.notifyBeep |
| Values: | true, false |
| Description | If set to true, printtool will beep when all print jobs are done. The default value is false (no beep). |

SEE ALSO

accept(1M), cancel(1), enable(1), filemgr(1), lp(1), lpadmin(1), lpfilter(1M), lpstat(1), lpsystem(1M), mailtool(1) xview(7)

Solaris User's Guide

"About Print Tool" in the Help Handbook available through the **Help** option on the Workspace menu.

DIAGNOSTICS

See **lpstat(1)** diagnostics.

BUGS

See **lp(1), lpstat(1)** bugs.

| | |
|--------------------|--|
| NAME | props – graphical interface to set OpenWindows properties |
| SYNOPSIS | props [-toolkitoption ...] [-init] |
| DESCRIPTION | <p>The props program provides a graphical interface to many properties of OpenWindows which are controlled via X resources. The resources props manipulates are stored in the file called .OWdefaults in the user's home directory. This enables props to ensure properties persist across window sessions.</p> <p>Upon window system startup, the .OWdefaults file is merged into the X resource manager as specified in the <i>openwin-sys</i> file. This occurs after xinit(1) loads \$OPENWINHOME/lib/Xdefaults or .Xdefaults in the user's home directory if it exists. Thus, the settings in .OWdefaults take precedence over the settings in a user's .Xdefaults file which in turn take precedence over the system defaults set in \$OPENWINHOME/lib/Xdefaults. When properties are changed via props, props updates .OWdefaults and the X resource manager.</p> <p>The resources handled by props consist of two resource components. The first component is OpenWindows. The second resource component is the name of the particular property being set. For example, the mouse acceleration is controlled by the resource OpenWindows.MouseAcceleration. The OpenWindows toolkits and window manager are designed to respond to the resources manipulated by props. Systems without the OpenWindows toolkits and window manager are not likely to be aware of the OpenWindows resources. Thus, if props is used in a non-OpenWindows environment, the system will probably ignore the settings made via props.</p> <p>Below is an explanation of the resources handled by props. The second component of each resource name is listed. For example, the name "WorkspaceColor" refers to the complete resource name OpenWindows.WorkspaceColor.</p> <p>BasicLocale (<i>locale name</i>) Specifies the basic locale in which the system is running. Permissible locale names are "C" (USA), "ja" (Japan), "ko" (Korea), "zh" (PRC), "zh_TW" (ROC), "de" (Germany), "it" (Italy), "fr" (France), and "sv" (Sweden). Default value: C Properties category: Locale Category control: Basic Locale The default specifies the U.S. locale and the English language.</p> <p>Beep (<i>enumeration</i>) Specifies the circumstances under which olwm(1) should beep. Permissible values are the strings "always", "never", and "notices". The string "never" means that olwm should never beep, "notices" means that olwm should beep only when a notice appears, and "always" means that olwm will beep whenever it is appropriate. Default value: always Properties category: Miscellaneous Category control: Beep</p> |

BeepDuration (*integer*)

Specifies how long the keyboard beep should last in milliseconds.

Default value: 100

Properties category: Miscellaneous

Category control: Beep Duration

BoldFont (*font name*)

Specifies the default bold font used for captions and window titles.

Default value: `*-lucida sans-bold-r-*-*-120-*-*-*`

Properties category: Fonts

Category control: Typeface

The default specifies a bold weight, 12-point, regular Lucida Sans font.

DataBackground (*color*)

Specifies the color for the background of text windows such as those in mailtool, textedit, and the help window.

Default value: #ffffff

Properties category: Color

Category control: DATA AREAS Background

The default value specifies white.

DataForeground (*color*)

Specifies the color for the text in text windows such as those in mailtool, textedit, and the help window.

Default value: #000000

Properties category: Color

Category control: DATA AREAS Foreground

The default value specifies black.

DisplayLang (*locale name*)

Specifies the locale for the language used when displaying text. See the description of the **BasicLocale** resource for a list of permissible locale names.

Default value: C

Properties category: Locale

Category control: Display Locale

The default specifies the U.S. locale and the English language.

DragRightDistance (*integer*)

The number of pixels you must drag the mouse to the right in a menu item to bring up a sub-menu. The sub-menu always comes up when you move over the menu mark (the right-pointing triangle), regardless of the drag-right distance.

Default value: 100

Properties category: Menus

Category control: Drag-Right Distance

IconLocation (*enumeration*)

One of the words "top-lr", "top-rl", "bottom-lr", "bottom-rl", "left-tb", "left-bt", "right-tb", or "right-bt". These specify that icons should be arranged along a particular edge of the screen, ordered from left to right or top to bottom as

appropriate. The words "top", "bottom", "left", and "right" are synonyms for "top-lr", "bottom-lr", "left-tb", and "right-tb", respectively. These synonyms are used by **props**.

Default value: bottom

Properties category: Miscellaneous

Category control: Icon Location

InputLang (*locale name*)

Specifies the language expected to be typed from the keyboard. See the description of the **BasicLocale** resource for a list of permissible locale names.

Default value: C

Properties category: Locale

Category control: Input Locale

The default specifies the U.S. locale and the English language.

KeyboardCommands (*enumeration*)

Permissible values for this resource are "Basic" and "Full". In **Full** mode, all OPEN LOOK Mouseless commands implemented by the window manager are active. See the section on Key Binding in the **olwm(1)** man page for further information. In **Basic** mode, the keys active are Open, Front, Help, and the colormap keys.

Default value: Basic

Properties category: Keyboard

Category control: Keyboard Mouse Equivalents

KeyClick (*boolean*)

Specifies whether keys click when pressed.

Default value: false

Properties category: Keyboard

Category control: Other Options, Key Click

KeyRepeat (*boolean*)

Specifies whether repeating keys repeat when held down.

Default value: true

Properties category: Keyboard

Category control: Other Options, Key Repeat

MenuAccelerators (*boolean*)

Specifies whether menu accelerators are enabled for menus in applications.

Default value: true

Properties category: Keyboard

Category control: Keyboard Menu Equivalents

MonospaceFont (*font name*)

Specifies the default monospace font used for text editors and other applications which require a monospace font.

Default value: `--lucida sans typewriter-medium-r-*-*--120-*-*-*-*--*`

Properties category: Fonts

Category control: Typeface

The default specifies a medium weight, 12-point, regular Lucida Sans Typewriter font.

MouseAcceleration (*integer*)

Specifies a multiplier. The mouse pointer will go this many times faster when it moves more than the number of pixels specified by the **MouseThreshold** resource in a short time.

Default value: 2

Properties category: Mouse

Category control: Mouse Acceleration

MouseThreshold (*integer*)

Specifies the number of pixels which the mouse must move in a short time for the mouse acceleration to be applied.

Default value: 15

Properties category: Mouse

Category control: Mouse Threshold

MultiClickTimeout (*integer*)

The time, in tenths of a second, that differentiates a double-click from two single clicks.

Default value: 5

Properties category: Mouse

Category control: Multi-Click Interval

NumericFormat

Specifies how commas and periods are used in numbers. See the description of the **BasicLocale** resource for a list of permissible locale names.

Default value: C

Properties category: Locale

Category control: Numeric Format

The default specifies the U.S. locale, a numeric format where commas are used to indicate thousands and a period is used to indicate where the fractional part of the number begins. For example, one thousand and a half is "1,000.5".

PointerMapping (*enumeration*)

Specifies a "left" or "right" handed mapping of the mouse buttons. For a 3-button mouse, "right" means button 1 is SELECT, button 2 is ADJUST, and button 3 is MENU. A value of "left" means button 1 is MENU, button 2 is ADJUST, and button 3 is SELECT. For mice with more or less than 3 buttons, the sense of the buttons is reversed as for the 3-button mouse.

Default value: right

Properties category: Mouse

Category control: Mouse Button Order

PopupJumpCursor (*boolean*)

Specifies whether to warp the cursor to popup windows.

Default value: true

Properties category: Mouse

Category control: Pointer Jumping, Pop-Up Windows**RegularFont** (*font name*)

Specifies the default font used by the system in general. This font is used for such text as that in buttons and non-bold labels.

Default value: `*-lucida sans-medium-r-*-*-120-*-*-*`

Properties category: Fonts

Category control: Typeface

The default specifies a medium weight, 12-point, regular Lucida Sans font.

Scale (*enumeration*)

Specifies the desktop scale. The scale value indicates the point size of the OPEN LOOK Glyph font and the text fonts used by OpenWindows. Permissible values are "small", "medium", "large", and "extra_large". These correspond to 10, 12, 14, and 19 point fonts, respectively.

Default value: medium

Properties category: Fonts

Category control: Scale

The default value specifies 12 point fonts.

ScreenSaver.IdleTime (*integer*)

Specifies the number of minutes of idle time required before the screen saver comes on. The **ScreenSaver.OnOff** resource must be set to "auto" for the screen-saver to activate. *Default value: 10.*

ScreenSaver.OnOff (*enumeration*)

Specifies whether the screensaver is off or come on automatically. The permissible values are "off" and "auto", respectively.

Default value: off

Properties category: Miscellaneous

Category control: Screen Saver

ScrollbarJumpCursor (*boolean*)

Specifies whether to warp the cursor to follow the scrollbar elevator.

Default value: true

Properties category: Mouse

Category control: Pointer Jumping, Scrollbars

ScrollbarPlacement (*enumeration*)

Specifies on which side of windows to place verticle scrollbars. Permissible values are "left" and "right".

Default value: right

Properties category: Miscellaneous

Category control: Scrollbar Placement

SelectDisplaysMenu (*boolean*)

If true, pressing the SELECT mouse button will bring up a menu item's sub-menu (if any) instead of executing the sub-menu's default action.

Default value: true

Properties category: Menus

Category control: Left Mouse Press**SetInput** (*enumeration*)

This controls the input focus mode. If the value is "select", it means click-to-focus. If the value is "followmouse", it means focus-follows-mouse.

Default value: select

Properties category: Miscellaneous

Category control: Set Active Window

TimeFormat

Specifies the locale for the date and time format. Date format is mm/dd/yy or dd/mm/yy and the time format is 12- or 24-hour clock. See the description of the **BasicLocale** resource for a list of permissible locale names.

Default value: C

Properties category: Locale

Category control: Time Format

The default specifies the U.S. locale which has date format mm/dd/yy and 12-hour clock time format.

WindowColor (*color*)

Specifies the color of windows. This is the "BG1" color for 3D OPEN LOOK. It is used for the backgrounds of windows, menus, and notices. The 3D effect is achieved by using highlight and shadow colors derived from this color.

Default value: #cccccc

Properties category: Color

Category control: WINDOWS Background

The default specifies a 20% gray value.

WindowForeground (*color*)

Specifies the color for foreground components of windows such as captions and text in buttons and footers.

Default value: #000000

Properties category: Color

Category control: WINDOWS Foreground

The default value specifies black.

WindowMenuAccelerators (*boolean*)

Specifies whether menu accelerators are enabled for window menus.

Default value: true

Properties category: Keyboard

Category control: Keyboard Menu Equivalent

WorkspaceBitmapBg (*color*)

Specifies the color for the background of the workspace bitmap which **olwm** tiles on the root window if the **WorkspaceStyle** resource is set to "tilebitmap".

Default value: #ffffff

Properties category: Color

Category control: DATA AREAS Background

The default value specifies white.

WorkspaceBitmapFg (*color*)

Specifies the color for the foreground of the workspace bitmap which **olwm** tiles on the root window if the **WorkspaceStyle** resource is set to "tilebitmap".

Default value: #000000

Properties category: Color

Category control: DATA AREAS Foreground

The default value specifies black.

WorkspaceColor (*color*)

Specifies the color for the workspace (root window). On startup, **olwm** will set the root window's background color to the color specified by this resource if the **WorkspaceStyle** resource is set to "paintcolor", and it will restore the default background on shutdown.

Default value: #40a0c0

Properties category: Color

Category control: WORKSPACE Background

The default value specifies a light blue color.

WorkspaceStyle (*enumeration*)

Specifies how **olwm** decorates the workspace (root window). Permissible values are "paintcolor" and "tilebitmap". If the value is "paintcolor", **olwm** paints the workspace with the color indicated by the **WorkspaceColor** resource. If the value is "tilebitmap", **olwm** tiles the bitmap indicated by the **WorkspaceBitmapFile** resource using the colors specified by the **WorkspaceBitmapFg** and **WorkspaceBitmapBg** resources.

Default value: paintcolor

Properties category: Color

Category control: Pattern

The default specifies to use the color selected by **WorkspaceColor**.

OPTIONS

The **props** program accepts all standard command line options accepted by the OLIT toolkit. In addition, it accepts the following.

-init This is the way **props** is started by **xinit** upon window system initialization. This is done to ensure the state of the following window server attributes are set in accordance with the resources in the **.OWdefaults** file in the user's home directory: key click, key repeat, beep duration, screen saver activation and timeout. When **props** is started with this option, it checks and resets the server state for these settings if necessary and exits. This option is only meant to be used upon system startup.

FILES**\$HOME/.OWdefaults**

Contains the resource name-value pairs written by the **props** program.

\$HOME/.openwin-menu-programs

Contains the user's custom programs menu.

\$HOME/.openwin-menu-programs.BAK

Contains a backup version of **\$HOME/.openwin-menu-programs**.

\$HOME/.Xdefaults Contains the user's resource name-value pairs as set outside of **props**.

/usr/openwin/lib/Xdefaults Contains the system's default resource name-value pairs.

/usr/openwin/lib/app-defaults/Props Contains strings and layout information for **props**.

/usr/openwin/lib/help/props.info Contains help text for **props**.

/usr/openwin/lib/openwin-sys Starts **props** in initialization mode upon window system startup.

/usr/openwin/bin/props The executable **props** program.

/usr/openwin/etc/workspace/patterns/*.xbm Workspace bitmap files.

/usr/openwin/etc/workspace/patterns/attributes Workspace bitmap colors.

/tmp/OWtemp Temporary resource storage file.

SEE ALSO

olwm(1) xinit(1) xmodmap(1) xrdb(1) xset(1) xview(7)

| | |
|--------------------|---|
| NAME | pswrap – creates C procedures from segments of PostScript language code |
| SYNOPSIS | pswrap [-apr] [-o <i>outputCfile</i>] [-h <i>outputHfile</i>] [-s <i>maxstring</i>] <i>inputfile</i> |
| DESCRIPTION | <p>pswrap reads input from <i>inputfile</i> and creates C-callable procedures, known as wraps, that send PostScript language code to the PostScript interpreter. <i>inputfile</i> contains segments of PostScript language code wrapped with a C-like procedure syntax.</p> <p>Wraps are the most efficient way for an application to communicate with the PostScript interpreter. For complete documentation of pswrap and the language it accepts, see "pswrap Reference Manual" in <i>Programming the Display PostScript System with X</i>.</p> |
| OPTIONS | <p><i>inputfile</i></p> <p>A file that contains one or more wrap definitions. pswrap transforms the definitions in <i>inputfile</i> into C procedures. If no input file is specified, the standard input (which can be redirected from a file or pipe) is used. The input file can include text other than wrap definitions. pswrap converts wrap definitions to C procedures and passes the other text through unchanged. Therefore, it is possible to intersperse C-language source code with wrap definitions in the input file.</p> <p><i>Note:</i> Although C code is allowed in a pswrap input file, it is not allowed within a wrap body. In particular, no CPP macros (for example, #define) are allowed inside a wrap.</p> <p>-a Generates ANSI C procedure prototypes for procedure definitions in <i>outputCfile</i> and, optionally, <i>outputHfile</i>. The -a option allows compilers that recognize the ANSI C standard to do more complete type checking of parameters. The -a option also causes pswrap to generate const declarations.</p> <p><i>Note:</i> ANSI C procedure prototype syntax is not recognized by most non-ANSI C compilers, including many compilers based on the Portable C Compiler. Use the -a option only in conjunction with a compiler that conforms to the ANSI C Standard.</p> <p>-h <i>outputHFile</i></p> <p>Generates a header file that contains extern declarations for non-static wraps. This file can be used in #include statements in modules that use wraps. If the -a option is specified, the declarations in the header file are ANSI C procedure prototypes. If the -h option is omitted, a header file is not produced.</p> <p>-o <i>outputCFile</i></p> <p>Specifies the file to which the generated wraps and passed-through text are written. If omitted, the standard output is used. If the -a option is also specified, the procedure definitions generated by pswrap are in ANSI C procedure prototype syntax.</p> <p>-p Specifies that strings passed by wraps are padded so that each data object begins on a long-word (4-byte) boundary. This option allows wraps to run on architectures that restrict data alignment to 4-byte boundaries and improves</p> |

performance on some other architectures.

- r** Generates reentrant code for wraps shared by more than one process (as in shared libraries). Reentrant code can be called recursively or by more than one thread. The **-r** option causes **pswrap** to generate extra code, so use it only when necessary.
- s maxstring** Sets the maximum allowable length of a PostScript string object or hexadecimal string object in the wrap body input. A syntax error is reported if a string is not terminated with **)** or **>** within *maxstring* characters. *maxstring* cannot be set lower than 80; the default is 200.

SEE ALSO *Programming the Display PostScript System with X* (Addison-Wesley Publishing Company, Inc., 1993).

AUTHOR Adobe Systems Incorporated

NOTES PostScript and Display PostScript are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions.
Copyright (c) 1988-1991 Adobe Systems Incorporated. All rights reserved.

| | |
|--------------------|---|
| NAME | puzzle – 15-puzzle game for X |
| SYNOPSIS | puzzle [-display <i>display</i>] [-geometry <i>geometry</i>] [-size <i>WIDTH</i> x <i>HEIGHT</i>] [-speed <i>num</i>] [-picture <i>filename</i>] [-colormap] |
| OPTIONS | <p>-display <i>display</i> This option specifies the display to use; see X11(7).</p> <p>-geometry <i>geometry</i> This option specifies the size and position of the puzzle window; see X11(7).</p> <p>-size <i>WIDTH</i> x <i>HEIGHT</i> This option specifies the size of the puzzle in squares.</p> <p>-speed <i>num</i> This option specifies the speed in tiles per second for moving tiles around.</p> <p>-picture <i>filename</i> This option specifies an image file containing the picture to use on the tiles. Try “mandrill.cm.” This only works on 8-bit pseudo-color screens.</p> <p>-colormap This option indicates that the program should create its own colormap for the picture option.</p> |
| DESCRIPTION | Puzzle with no arguments plays a 4x4 15-puzzle. The control bar has two boxes in it. Clicking in the left box scrambles the puzzle. Clicking in the right box solves the puzzle. Clicking the middle button anywhere else in the control bar causes puzzle to exit. Clicking in the tiled region moves the empty spot to that location if the region you click in is in the same row or column as the empty slot. |
| SEE ALSO | X11(7) |
| BUGS | The picture option should work on a wider variety of screens. |
| COPYRIGHT | Copyright 1988, Don Bennett. |
| AUTHOR | Don Bennett, HP Labs |

| | |
|--------------------|---|
| NAME | ras2ps – converts a Sun RasterFile to a PostScript file |
| SYNOPSIS | ras2ps [-x <i>xoffset</i>] [-y <i>yoffset</i>] [-X <i>xscale</i>] [-Y <i>yscale</i>] [-w <i>width</i>] [-h <i>height</i>] [-r <i>rotation</i>] [-i] [-C] [-l] [-n] [-v] [-q] [<i>rasterfile</i> -] [<i>psfile</i>] |
| DESCRIPTION | ras2ps converts a Sun RasterFile to a PostScript file. If both filenames are missing, the rasterfile is read from stdin and the PostScript is written to stdout . If there is only one filename, then it is interpreted as the rasterfile and is opened for input. To have a named output PostScript file and still read the rasterfile from stdin , use a dash (-) in place of the input filename. |
| OPTIONS | <p>-x <i>xoffset</i> Set the amount of space to the left of the image to be <i>xoffset</i> inches. The default is a quarter inch to keep the image in the imagable area of the printer.</p> <p>-y <i>yoffset</i> Set the amount of space below the image to <i>yoffset</i> inches. The default is a quarter inch same as above.</p> <p>-X <i>xscale</i> Multiply the width of the image by <i>xscale</i>. This is used to stretch or shrink an image along the X axis.</p> <p>-Y <i>yscale</i> Multiply the height of the image by <i>yscale</i>. This is used to stretch or shrink an image along the Y axis.</p> <p>-w <i>width</i> Set the max <i>width</i> in inches.</p> <p>-h <i>height</i> Set the max <i>height</i> in inches. The <i>-w</i> and <i>-h</i> options set the desired width and height of the <i>output</i> image in inches. Default height and width are the source image dimensions at 300 dpi. ras2ps will expand or shrink the image to fit these dimensions, while still preserving the image scale values. The closest fit within the boundaries will be used. Note that width and height arguments do not stretch or shrink the image, but only set its limits. At least <i>one</i> of width or height is guaranteed to be satisfied.</p> <p>-r <i>n</i> Rotate the image by <i>n</i> degrees, counterclockwise. The origin of rotation is the lower left corner of the image at the point specified by the <i>-x</i> and <i>-y</i> options.</p> <p>-i Invert the image. This will reverse black and white on a monochrome image and is of limited usefulness on color images.</p> <p>-C Output 8 and 24 bit images as color PostScript using the <i>colorimage</i> operator as supported by printers such as the <i>QMS ColorPS 800</i>.</p> <p>-l Orient the image in landscape mode, which puts the origin at the lower right corner of the page and rotates the image 90 degrees. All arguments follow this new orientation. The default is Portrait mode.</p> |

- n Do not include the PostScript operator 'showpage' in the output. This is for backward compatibility with programs which do not override 'showpage' as the EPSF spec advises.
- v Verbose mode will print information as it processes the image. (The default is to be silent.)
- q Query (prints list of options)

SEE ALSO [lp\(1\)](#), [24to8\(1\)](#)

| | |
|--------------------|---|
| NAME | rash – Sun Raster to PostScript translator |
| SYNOPSIS | rash [-2] [-n x m] [-s width height [[-H height] [-S height] [-W width] [-l xpos ypos] [-c xpos ypos] [-R angle] [-e] [-i] [-m] [-n] [-p prolog] [-r] [-wrapper] [<i>filename...</i>] |
| DESCRIPTION | <p>rash converts a Sun Raster file into a PostScript file, using a PostScript "wrapper" to surround the raster input. By default, it creates a color PostScript file if given a color image and a monochrome PostScript file if given a monochrome image.</p> <p>By default, rash centers the image on the page and orient and scale it to fill up as much of the printable area as possible while preserving the image's aspect ratio.</p> <p>All the options regarding the size of the raster file take a number followed by a unit, where a unit is one of in, cm, mm, or point. measurements and units can be combined; it is possible to have a length of "1 in .5 cm", for example. The default measurement unit is points.</p> <p>rash is most often used with the PreLinn filter manager, part of the NeWSprint package.</p> |
| OPTIONS | <p>-2 Scale the image by a factor of 2. Each pixel in the rasterfile is printed as 4 pixels (2 x 2) on the output device.</p> <p>-n x m Scale the image by a factor of <i>n</i> in the horizontal direction and <i>m</i> in the vertical direction. If only one number is specified, both dimensions are scaled evenly.</p> <p>-s width height Scale the image to the specified width and height. This option does not preserve the original raster's aspect ratio.</p> <p>-H height Scale the image (preserving aspect ratio) to the specified height.</p> <p>-S height Same as -H.</p> <p>-W width Scale the image (preserving aspect ratio) to the specified width.</p> <p>-l xpos ypos Locate the lower left corner of the image <i>xpos</i> and <i>ypos</i> units in the x and y direction. The origin is at the lower left corner of the page.</p> <p>-c xpos ypos Locate the center of the image <i>xpos</i> and <i>ypos</i> units from the center of the page.</p> <p>-R angle Rotate the image to the specified angle, in degrees. (Probably most useful with the -c option.)</p> <p>-e Output Encapsulated PostScript (EPSF). The -c, -2, -n x m, and -i options are illegal when producing EPSF, the -n option is ignored, and a size argument (one of -s, -H, -S, or -W) is required.</p> <p>-i Reduce the scale, if necessary, to an integral number of pixels output for each pixel input. This can improve output quality in some situations, notably when printing a screen dump which includes icons.</p> <p>-m If the input is a color rasterfile, convert it to 8-bit gray scale.</p> |

- n** Do not output a **showpage** command at the end of the file, for PostScript files that will be imported into other documents. If you are converting a rasterfile to PostScript for inclusion in another document, the **-e** option (EPSF) may be a better idea.
- p** *prolog* Output the PostScript to the specified file rather than to the standard output.
- r** Print in landscape mode
- w** *wrapper* Use the specified file as the PostScript wrapper.

WARNINGS

If you are sending a color rasterfile to a PostScript printer that does not implement the **colorimage** operator, you should override printing in color by using the **-m** option. Sun Raster format uses 1 for black values and 0 for white. PostScript uses the reverse values. **rash** converts all values to PostScript equivalents.

SEE ALSO

rasterfile(5),
PostScript Language Reference Manual
PostScript Document Structuring Conventions
Encapsulated PostScript Files
PreLinn User's Guide
NeWSprint Installation and Administration Guide

BUGS

rash may not output "true" EPSF; it relies on the wrapper file for the proper structuring.

| | |
|--------------------|---|
| NAME | rasterfile – Sun’s file format for raster images |
| SYNOPSIS | #include <rasterfile.h> |
| DESCRIPTION | <p>A rasterfile is composed of three parts: first, a header containing 8 integers; second, a (possibly empty) set of colormap values; and third, the pixel image, stored a line at a time, in increasing y order. The image is layed out in the file as in a memory pixrect. Each line of the image is rounded up to the nearest 16 bits.</p> <p>The header is defined by the following structure:</p> <pre> struct rasterfile { int ras_magic; int ras_width; int ras_height; int ras_depth; int ras_length; int ras_type; int ras_maptype; int ras_maplength; }; </pre> <p>The <i>ras_magic</i> field always contains the following constant:</p> <pre> #define RAS_MAGIC 0x59a66a95 </pre> <p>The <i>ras_width</i>, <i>ras_height</i>, and <i>ras_depth</i> fields contain the image’s width and height in pixels, and its depth in bits per pixel, respectively. The depth is either 1 or 8, corresponding to standard frame buffer depths. The <i>ras_length</i> field contains the length in bytes of the image data. For an unencoded image, this number is computable from the <i>ras_width</i>, <i>ras_height</i>, and <i>ras_depth</i> fields, but for an encoded image it must be explicitly stored in order to be available without decoding the image itself. Note: the length of the header and of the (possibly empty) colormap values are not included in the value of the <i>ras_length</i> field; it is only the image data length. For historical reasons, files of type RT_OLD will usually have a 0 in the <i>ras_length</i> field, and software expecting to encounter such files should be prepared to compute the actual image data length if needed. The <i>ras_maptype</i> and <i>ras_maplength</i> fields contain the type and length in bytes of the colormap values, respectively. If <i>ras_maptype</i> is not RMT_NONE and the <i>ras_maplength</i> is not 0, then the colormap values are the <i>ras_maplength</i> bytes immediately after the header. These values are either uninterpreted bytes (usually with the <i>ras_maptype</i> set to RMT_RAW) or the equal length red, green and blue vectors, in that order (when the <i>ras_maptype</i> is RMT_EQUAL_RGB). In the latter case, the <i>ras_maplength</i> must be three times the size in bytes of any one of the vectors.</p> |

| | |
|--------------------|--|
| NAME | realxfishdb, fish_props – Display a fishtank on the root window of an X11 server. |
| SYNOPSIS | realxfishdb [-fnn] [-bn] [-rn.n] [-in.n] [-d] [-s] fish_props |
| DESCRIPTION | realxfishdb displays a fishtank on the root window complete with swimming fish and bubbles, all in living color. This demo runs on Open Windows X11/NeWS servers v2 and newer. Currently only 8 bit color is supported. The fish_props program allows the user to preset various options for realxfishdb . |
| OPTIONS | <p>-fnn The number of fish. The default is 2. This option is ignored if you have used the fish properties sheet and a <code>~/fishrc</code> file can be found.</p> <p>-bn The number of bubbles. The suggested number is 0 since they take a significant amount of CPU power. The default is 0.</p> <p>-rn.n Fractions of a second between refresh updates. On an SS1 with about 8 fish and no hardware multibuffering support, setting this to 0.3 leaves you enough CPU power to continue to edit or read mail. Default is 1.0 (which corresponds to 100 on the property sheet). This option is ignored if you have used the fish properties sheet and a <code>~/fishrc</code> file can be found.</p> <p>-in.n Maximum fish increment. For those who like smoothly moving fish, set this value to a low value such as 0.2. The default is 1.0 (which corresponds to 100 on the property sheet). This option is ignored if you have used the fish properties sheet and a <code>~/fishrc</code> file can be found.</p> <p>-d Use the X11 Multibuffering Extension in OWv3 and newer Open Windows servers. This is not recommended unless you have hardware support for multibuffering since it further drains memory and CPU power. The default is off. This option is ignored if you have used the fish properties sheet and a <code>~/fishrc</code> file can be found.</p> <p>-s Secure mode. The default is off. This mode covers all windows on the screen and runs in a secure mode. Windows can't even be popped up to stop the program in this mode.</p> <p>fish_props is a fish properties sheet which allows the user to select the fish used in the program, the display mode, the refresh rate, and the maximum fish increment. After making selections press the apply button to save selections in the <code>~/fishrc</code> file. realxfishdb attempts to read this file when it starts.</p> |
| ENVIRONMENT | <p>There are image files in the <code>\$OPENWINHOME/share/images/fish</code> directory which are required to run realxfishdb.</p> <p>FISHHOME Indicates an alternate path for fish_props and realxfishdb to search for the fish files otherwise if they can't be found in the <code>\$OPENWINHOME</code> directory the final path attempted is the <code>./fish</code> directory relative to where the executable lives.</p> |

| | |
|--------------|---|
| FILES | \$HOME/.fishrc Contains the options set by running fish_props . realxfishdb attempts to read this file upon startup. |
| | \$OPENWINHOME/share/images/fish/*.im8 Image files necessary to run fish_props and realxfishdb . |
| NOTES | Be patient when starting this program, it processes a lot of pixels before starting up. Also, it runs on the root window so don't be alarmed when windows temporarily disappear. If you would like your very own copy you will need the fish_props and realxfishdb executables and all the files from the ./fish subdirectory. This is an adaptation of an existing X11 application called xaqua of unknown origin. |
| BUGS | Both fish_props and realxfishdb dump core if the ./fish directory does not contain the required image files. |

| | |
|--------------------|--|
| NAME | redxblue – swap red and blue for a 24 or 32 bit rasterfile |
| SYNOPSIS | redxblue [-v] [-q] [<i>inrasf</i>] [<i>outrasf</i>] |
| DESCRIPTION | <p>redxblue converts an old-style 24 or 32 bit rasterfile into the newer, Sun-standard format. The old format had the byte ordering RGB for 24-bit rasterfiles and XRGB for 32-bit rasterfiles. The new format has the byte ordering XBGR for both 24-bit and 32-bit rasterfiles. X stands for undefined byte value.</p> <p>The conversion is performed simply by swapping the red and blue bytes.</p> <p>It is also possible to use this filter to swap 'red' and 'blue' bytes in any 32-bit rasterfile.</p> |
| OPTIONS | <p>-v Verbose mode will print information as it processes the image. (The default is to be silent.)</p> <p>-q Query (prints list of options)</p> |
| SEE ALSO | 24to8(1) |

| | |
|--------------------|--|
| NAME | resize – utility to set TERM and terminal settings to current window size |
| SYNOPSIS | resize [-u -c] [-s [row col]] |
| DESCRIPTION | resize outputs a shell command for setting the TERM environment variable to indicate the current size of the xterm (1) window from which the command is run. For this output to take effect, resize must either be evaluated as part of the command line (usually done with a shell alias or function) or else redirected to a file which can then be read. |
| OPTIONS | The following options may be used with resize : <ul style="list-style-type: none"> -u This option indicates that Bourne shell commands should be generated even if the user's current shell isn't /bin/sh. -c This option indicates that C shell commands should be generated even if the user's current shell isn't /bin/csh. -s [row col] <ul style="list-style-type: none"> This option indicates that that Sun console escape sequences will be used instead of the special xterm escape code. If <i>row</i> and <i>col</i> are given, resize will ask the xterm to resize itself. However, the window manager may choose to disallow the change. |
| EXAMPLES | From the C shell (usually known as /bin/csh), the following alias could be defined in the user's .cshrc : <pre style="margin-left: 2em;">example% alias rs 'set noglob; `eval resize`'</pre> <p>After resizing the window, the user would type:</p> <pre style="margin-left: 2em;">example% rs</pre> <p>Users of versions of the Bourne shell (usually known as /bin/sh) that don't have command functions will need to send the output to a temporary file and then read it back in with the "." command:</p> <pre style="margin-left: 2em;">\$ resize >/tmp/out \$./tmp/out</pre> |
| FILES | /usr/share/lib/termcap /usr/share/lib/terminfo/?/* ~/cshrc |
| SEE ALSO | csh(1) , tset(1B) , xterm(1) |

AUTHORS

Mark Vandevoorde (MIT-Athena), Edward Moy (Berkeley)
Copyright (c) 1984, 1985 by Massachusetts Institute of Technology.
See **X11(7)** for a complete copyright notice.

BUGS

The **-u** or **-c** must appear to the left of **-s** if both are specified.
There should be some global notion of display size; **terminfo** needs to be rethought in the context of window systems. (Fixed in 4.3BSD, and Ultrix-32 1.2)

| | | |
|--------------------|---|---|
| NAME | rgb – build the color name database | |
| SYNOPSIS | rgb [<i>dbname</i>] | |
| DESCRIPTION | <p>rgb reads from standard input lines of the form:</p> <p style="padding-left: 40px;"><i>red green blue name</i></p> <p>where <i>red</i> / <i>green</i> / <i>blue</i> are decimal values between the range 0 to 255, and <i>name</i> is a description of the color. rgb then builds a color name database in dbm format. The color name database provides a mapping between ASCII color names and RGB color values. It is useful for increasing the portability of color programs. The input source for the database is in /usr/openwin/lib/rgb.txt. rgb.txt is compiled into the dbm files rgb.dir and rgb.pag. When the server first starts up, it consults the contents of these files to build an internal representation of their contents. This internal representation is consulted to map color names to color values.</p> <p>rgb uses the default color name database of /usr/openwin/lib/X11/rgb.txt.</p> | |
| OPTIONS | <i>dbname</i> | Color name database. |
| EXAMPLES | <p>example% rgb rgb < rgb.txt</p> <p>example% cat /usr/openwin/lib/rgb.txt</p> | |
| FILES | /usr/openwin/lib/rgb.txt | color name database source. Maps color names to RGB color values. |
| | /usr/openwin/lib/rgb.dir | dbm file containing color name to RGB mapping. |
| | /usr/openwin/lib/rgb.pag | dbm file containing the color name to RGB mapping. |
| SEE ALSO | bldrgb(1) , cat(1) , dbm(3) | |

| | |
|--------------------|---|
| NAME | rpc.cmsd – calendar manager service daemon |
| SYNOPSIS | <code>/usr/openwin/bin/rpc.cmsd [-d] [-s]</code> |
| DESCRIPTION | rpc.cmsd is a small database manager for appointment and resource-scheduling data. It's primary client is Calendar Manager, a productivity tool included with OpenWindows. rpc.cmsd is normally invoked by inetd (1M) when a Calendar Manager request is recieved. |
| OPTIONS | -d Enables debugging output. -s Runs rpc.cmsd in the foreground. This option should be used when rpc.cmsd is invoked manually for debugging purposes. |
| FILES | <code>/usr/spool/calendar/callog.\$USER</code> <code>/etc/inetd.conf</code> |

| | |
|--------------------|---|
| NAME | sessreg – manage utmp/wtmp entries for non-init clients |
| SYNOPSIS | sessreg [-w <i>wtmp-file</i>] [-u <i>utmp-file</i>] [-l <i>line-name</i>] [-h <i>host-name</i>] [-s <i>slot-number</i>] [-x <i>Xservers-file</i>] [-t <i>ttys-file</i>] [-a] [-d] <i>user-name</i> |
| DESCRIPTION | <p>Sessreg is a simple program for managing utmp/wtmp entries for xdm(1) sessions. System V has a better interface to /etc/utmp than BSD; it dynamically allocates entries in the file, instead of writing them at fixed positions indexed by position in /etc/ttys.</p> <p>To manage BSD-style utmp files, sessreg has two strategies. In conjunction with xdm(1), the -x option counts the number of lines in /etc/ttys and then adds to that the number of the line in the Xservers file which specifies the display. The display name must be specified as the "line-name" using the -l option. This sum is used as the "slot-number" in /etc/utmp that this entry will be written at. In the more general case, the -s option specifies the slot-number directly. If for some strange reason your system uses a file other than /etc/ttys to manage init, the -t option can direct sessreg to look elsewhere for a count of terminal sessions.</p> <p>Conversely, System V managers will not ever need to use these options (-x, -s and -t). To make the program easier to document and explain, sessreg accepts the BSD-specific flags in the System V environment and ignores them.</p> <p>BSD also has a host-name field in the utmp file which doesn't exist in System V. This option is also ignored by the System V version of sessreg.</p> |
| USAGE | <p>In Xstartup, place a call like:</p> <pre>sessreg -a -l \$DISPLAY -x /usr/openwin/lib/X11/xdm/Xservers \$USER</pre> <p>and in Xreset:</p> <pre>sessreg -d -l \$DISPLAY -x /usr/openwin/lib/X11/xdm/Xservers \$USER</pre> |
| OPTIONS | <p>-w <i>wtmp-file</i> This specifies an alternate wtmp file, instead of /usr/adm/wtmp for BSD or /etc/wtmp for sysV. The special name "none" disables writing records to /usr/adm/wtmp.</p> <p>-u <i>utmp-file</i> This specifies an alternate utmp file, instead of "/etc/utmp". The special name "none" disables writing records to /etc/utmp.</p> <p>-l <i>line-name</i> This describes the "line" name of the entry. For terminal sessions, this is the final pathname segment of the terminal device filename (e.g. ttyd0). For X sessions, it should probably be the local display name given to the users session (e.g. :0). If none is specified, the terminal name will be determined with ttyname(3) and stripped of leading components.</p> <p>-h <i>host-name</i> This is set for BSD hosts to indicate that the session was initiated from a remote</p> |

host. In typical xdm usage, this options is not used.

-s *slot-number*

Each potential session has a unique slot number in BSD systems, most are identified by the position of the *line-name* in the */etc/ttys* file. This option overrides the default position determined with **ttyslot(3C)**. This option is inappropriate for use with xdm, the **-x** option is more useful.

-x *Xservers-file*

As X sessions are one-per-display, and each display is entered in this file, this options sets the *slot-number* to be the number of lines in the *ttys-file* plus the index into this file that the *line-name* is found.

-t *ttys-file*

This specifies an alternate file which the **-x** option will use to count the number of terminal sessions on a host.

-a This session should be added to utmp/wtmp.

-d This session should be deleted from utmp/wtmp. One of **-a/-d** must be specified.

SEE ALSO

xdm(1)

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Keith Packard, MIT X Consortium

| | | | | | | | | | |
|--------------------------------|---|---|----------|---------------------|---|-----------------------------|---|-----------------------------|--|
| NAME | shelltool – run a shell (or other program) in an OpenWindows terminal window | | | | | | | | |
| SYNOPSIS | shelltool [-C] [-B boldstyle] [-I command] [<i>generic-tool-arguments</i>] [<i>program</i> [<i>arguments</i>]] | | | | | | | | |
| DESCRIPTION | <p>shelltool is a standard <i>OpenWindows</i> facility for shells or other programs that use a standard tty-based interface.</p> <p>When invoked, shelltool runs a program, (usually a shell) in an interactive terminal emulator based on a tty subwindow. Keyboard input is passed to that program. In the OpenWindows version of shelltool, a restricted pop-up menu is available from the main display area that allows you to enable scrolling. Selecting the Enable Scrolling option from the restricted menu gives shelltool the full functionality of the cmdtool window, including a larger pop-up menu from which to select options. Selecting Disable Scrolling from the pop-up submenu will return shelltool to its original state.</p> | | | | | | | | |
| OPTIONS | <p>-C Redirect system console output to this shelltool.</p> <p>-B boldstyle Set the style for displaying bold text to <i>boldstyle</i>. <i>boldstyle</i> can be a string specifying one of the choices for the term.boldstyle default, see Defaults Options, below, or it may be a numerical value for one of those choices, from 0 to 8, corresponding to the placement of the choice in the list.</p> <p>-I command Pass <i>command</i> to the shell. SPACE characters within the command must be escaped.</p> <p><i>generic-tool-arguments</i> shelltool accepts the generic tool arguments listed in xview(7).</p> | | | | | | | | |
| USAGE | | | | | | | | | |
| .Xdefaults File Options | You can specify a number of defaults using the options in the .Xdefaults file that affect the behavior of shelltool . The ones of interest are those that begin with text , term , or keyboard . See xview(7) for more detailed information. | | | | | | | | |
| The Terminal Emulator | The tty subwindow is a terminal emulator. Whenever a tty subwindow is created, the startup file ~.ttyswrc is read for initialization parameters that are specific to the tty subwindow. | | | | | | | | |
| The .ttyswrc File | <p>The command format for this file is:</p> <table border="0"> <tr> <td style="padding-right: 2em;">#</td> <td>Comment.</td> </tr> <tr> <td style="padding-right: 2em;">set variable</td> <td>Turn on the specified <i>variable</i>.</td> </tr> <tr> <td style="padding-right: 2em;">map<i>i</i> key text</td> <td>When <i>key</i> is typed pretend <i>text</i> was input.</td> </tr> <tr> <td style="padding-right: 2em;">map<i>o</i> key text</td> <td>When <i>key</i> is typed pretend <i>text</i> was output.</td> </tr> </table> <p>The only currently defined <i>variable</i> is pagemode. <i>key</i> is one of L1-L15, F1-F15, T1-T15, R1-R15, LEFT, or RIGHT on SPARC keyboards. On x86 keyboards <i>key</i> is one of F1-F12, End, PgDn, PgUp, HOME, LEFT, or RIGHT (see note below). <i>text</i> may contain escapes such</p> | # | Comment. | set variable | Turn on the specified <i>variable</i> . | map<i>i</i> key text | When <i>key</i> is typed pretend <i>text</i> was input. | map<i>o</i> key text | When <i>key</i> is typed pretend <i>text</i> was output. |
| # | Comment. | | | | | | | | |
| set variable | Turn on the specified <i>variable</i> . | | | | | | | | |
| map<i>i</i> key text | When <i>key</i> is typed pretend <i>text</i> was input. | | | | | | | | |
| map<i>o</i> key text | When <i>key</i> is typed pretend <i>text</i> was output. | | | | | | | | |

as `\E`, `\n`, `^X`, etc. (ESC, RETURN, and CTRL-X, respectively). See `curl_terminfo(3X)` for the format of the string escapes that are recognized. Note: `mapi` and `mapo` may be replaced by another keymapping mechanism in the future.

When using the default xserver keyboard tables, the keys L1, LEFT, RIGHT, BREAK, R8, R10, R12, and R14 on a SPARC keyboard cannot be mapped in this way; they send special values to the tty subwindow. Also, when using the default xserver keyboard tables, L1-L10 are now used by XView. Likewise, on an x86 keyboard, the keys LEFT, RIGHT, UP, and DOWN cannot be mapped this way. See `kbd(1)` for more information on how to change the behavior of the keyboard.

It is possible to have terminal-based special escape sequences. These escape sequences may also be sent by typing a key appropriately mapped using the `mapo` function described above. The following functions pertain to the tool in which the tty subwindow resides, not the tty subwindow itself.

| | |
|------------------------------|---|
| <code>\E[1t</code> | – open |
| <code>\E[2t</code> | – close (become iconic) |
| <code>\E[3t</code> | – move, with interactive feedback |
| <code>\E[3;TOP;LEFTt</code> | – move, to TOP LEFT (pixel coordinates) |
| <code>\E[4t</code> | – stretch, with interactive feedback |
| <code>\E[4;HT;WIDTHt</code> | – stretch, to HT WIDTH size (in pixels) |
| <code>\E[5t</code> | – front |
| <code>\E[6t</code> | – back |
| <code>\E[7t</code> | – refresh |
| <code>\E[8;ROWS;COLSt</code> | – stretch, to ROWS COLS size (in characters) |
| <code>\E[11t</code> | – report if open or iconic by sending <code>\E[1t</code> or <code>\E[2t</code> |
| <code>\E[13t</code> | – report position by sending <code>\E[3;TOP;LEFTt</code> |
| <code>\E[14t</code> | – report size in pixels by sending <code>\E[4;HT;WIDTHt</code> |
| <code>\E[18t</code> | – report size in characters by sending <code>\E[8;ROWS;COLSt</code> |
| <code>\E[20t</code> | – report icon label by sending <code>\E]Llabel\E\</code> |
| <code>\E[21t</code> | – report tool header by sending <code>\E]lheader\E\</code> |
| <code>\E]ltext\E\</code> | – set tool header to text |
| <code>\E]lfile\E\</code> | – set icon to the icon contained in file; file must be in <i>iconedit</i> output format |
| <code>\E]Llabel\E\</code> | – set icon label to label |
| <code>\E[>OPT;...h</code> | – turn SB OPT on (OPT = 1 => pagemode), for example, <code>\E[>1;3;4h</code> |
| <code>\E[>OPT;...k</code> | – report OPT; sends <code>\E[>OPTl</code> or <code>\E[>OPTH</code> for each OPT |
| <code>\E[>OPT;...l</code> | – turn OPT off (OPT = 1 => pagemode), for <code>.B \E[>1;3;</code> |

See the *Solaris User's Guide* for an example of using this facility.

shelltool Windows

The window created by `shelltool` is based on the `ttysw` package. This package provides a simple character-based terminal emulator interface. The user is given a prompt at which to type commands and pop-up menus from which to select command options.

shelltool windows support cursor motions, using the `/usr/share/lib/terminfo` entry called **sun-cmd**. Command windows automatically set the **TERM** environment variable to **sun-cmd**. So, if you **rlogin(1)** to a machine that does not have an entry for **sun-cmd** in its `/usr/share/lib/terminfo` file, the error message **'Type sun-cmd unknown'** results. To rectify this, type the command **'set TERM=sun'**. Programs written using the **curses(3X)** library packages will work in a command window, but programs hard-coded for **sun**-type terminals may not. When supporting a program that performs cursor motions, the command window automatically takes on the characteristics of a tty window (as with **shelltool**). When that program terminates or sleeps, the full command window functionality is restored.

The shelltool Menu

The **shelltool** window menu is called the **Term Pane** menu and contains the following options and their submenus:

Enable Page Mode

Enables page mode within **shelltool** .

Copy Places the highlighted text on the clipboard.

Paste Puts the contents of pointed to by the cursor.

Scrolling Enables scrolling within **shelltool** .

EXAMPLES

The following aliases can be put into your `~/.cshrc` file:

```
# dynamically set the name stripe of the tool:
alias header 'echo -n "\ E]I" \ !*\ E\ \ '
# dynamically set the label on the icon:
alias iheader 'echo -n "\ E]L" \ !*\ E\ \ '
# dynamically set the image on the icon:
alias icon 'echo -n "\ E]I" \ !*\ E\ \ '
```

FILES

```
~/ttypswrc
/usr/openwin/lib/ttypswrc
/usr/openwin/bin/shelltool
/usr/demo/*
```

SEE ALSO

cmdtool(1), **more(1)**, **xview(7)**, **rlogin(1)**, **kbd(1)**, **curs_terminfo(3X)**
Solaris User's Guide

| | | | | | | | |
|---------------------------------|---|---------------------------------|---|---------------------------------|--|---------------------------------|--|
| NAME | showrgb – display the color name database | | | | | | |
| SYNOPSIS | showrgb [dbname] | | | | | | |
| DESCRIPTION | <p>showrgb displays the contents of the dbm format color name database. The color name database provides a mapping between ASCII color names and RGB color values. It is useful for increasing the portability of color programs. The source of the database is in /usr/openwin/lib/X11/rgb.txt. rgb.txt is compiled into the dbm database files rgb.dir and rgb.pag. When the server first starts up, it consults the contents of these files to build an internal representation of their contents. This internal representation is consulted to map color names to color values. showrgb is useful in debugging color name databases to guarantee that the binary file is representative of the ASCII source file.</p> <p>showrgb uses the default color name database of /usr/openwin/lib/X11/rgb.txt.</p> | | | | | | |
| OPTIONS | dbname Color name database. | | | | | | |
| EXAMPLES | example% showrgb /usr/openwin/lib/rgb example% cat /usr/openwin/lib/rgb.txt | | | | | | |
| FILES | <table><tr><td>/usr/openwin/lib/rgb.txt</td><td>color name database source. Maps color names to RGB color values.</td></tr><tr><td>/usr/openwin/lib/rgb.dir</td><td>dbm file containing color name to RGB mapping.</td></tr><tr><td>/usr/openwin/lib/rgb.pag</td><td>dbm file containing color name to RGB mapping.</td></tr></table> | /usr/openwin/lib/rgb.txt | color name database source. Maps color names to RGB color values. | /usr/openwin/lib/rgb.dir | dbm file containing color name to RGB mapping. | /usr/openwin/lib/rgb.pag | dbm file containing color name to RGB mapping. |
| /usr/openwin/lib/rgb.txt | color name database source. Maps color names to RGB color values. | | | | | | |
| /usr/openwin/lib/rgb.dir | dbm file containing color name to RGB mapping. | | | | | | |
| /usr/openwin/lib/rgb.pag | dbm file containing color name to RGB mapping. | | | | | | |
| SEE ALSO | bldrgb(1) , cat(1) , rgb(1) , dbm(3) | | | | | | |

| | |
|--------------------|--|
| NAME | showsnf - print contents of an SNF file |
| SYNOPSIS | showsnf [-v] [-g] [-m] [-M] [-l] [-L] [-p#] [-u#] |
| DESCRIPTION | The showsnf utility displays the contents of font files in the Server Natural Format produced by bdftosnf . It is usually only to verify that a font file hasn't been corrupted or to convert the individual glyphs into arrays of characters for proofreading or for conversion to some other format. |
| OPTIONS | <p>-v This option indicates that character bearings and sizes should be printed.</p> <p>-g This option indicates that character glyph bitmaps should be printed.</p> <p>-m This option indicates that the bit order of the font is MSBFirst (most significant bit first).</p> <p>-l This option indicates that the bit order of the font is LSBFirst (least significant bit first).</p> <p>-M This option indicates that the byte order of the font is MSBFirst (most significant byte first).</p> <p>-L This option indicates that the byte order of the font is LSBFirst (least significant byte first).</p> <p>-p# This option specifies the glyph padding of the font (# is a number).</p> <p>-u# This option specifies the scanline unit of the font (# is a number).</p> |
| SEE ALSO | bdftosnf(1) |
| BUGS | There is no way to just print out a single glyph. |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |

| | |
|---------------------|--|
| NAME | snapshot – capture some or all of a screen image and save to a raster file |
| SYNOPSIS | snapshot [-d <i>default directory</i>] [-f <i>default filename</i>] [-g] [-l <i>filename</i>] [-n] [-v] |
| AVAILABILITY | This command is available with the OpenWindows environment. |
| DESCRIPTION | <p>snapshot is an OpenWindows XView utility that allows users to save images from their workstation screens into a raster file. The user may choose to take a picture of a single window, the entire screen, or any rectangular portion of the screen.</p> <p>snapshot can also be used to load and display Sun rasterfile or GIF formatted files. It is also capable of viewing images of a different depth than the screen, for example, displaying color images on a monochrome screen. The image is appropriately dithered before it is displayed. Compressed images will be unpacked and loaded automatically.</p> <p>Images can be saved to disk in Sun rasterfile format, or printed on a PostScript printer. There are various printer options (see below).</p> <p>If it also possible to drag and drop rasterfile or GIF images onto the snapshot drop zone and these will be loaded automatically. Similarly, it is possible to drag and drop from snapshot to another DeskSet application such as filemgr(1) or printtool(1).</p> |
| OPTIONS | <p>-d <i>default-directory</i> Name of the default directory to use for load and save operations.</p> <p>-f <i>default-filename</i> Name of the default filename to use for load and save operations.</p> <p>-g Display color images using a greyscale ramp.</p> <p>-l <i>filename</i> Name of the file to automatically load and view on startup.</p> <p>-n Automatically overwrite previously loaded images or snapshot without prompting the user for confirmation. Similarly, when saving images to disk, files will be automatically overwritten.</p> <p>-v Print the version number of this release of the snapshot program.</p> |
| USAGE | <p>snapshot operates using a combination of control panels with various buttons, choices and text fields to define the actions performed and the files used.</p> <p>Load... This button displays a control panel which allows the user to specify the name of the directory and the filename to use for image load operations. The default directory name is the current working directory, and the default filename is snapshot.rs. These text fields can be overwritten by the user.</p> <p>Save... This button displays a control panel which allows the user to specify the name of the directory and filename to use for saving snapped images. As with the load operation, the default directory name is the current working directory and the default filename is snapshot.rs, both of which can be adjusted by the user.</p> |

| | |
|--------------|--|
| Print | This button displays a menu with two options: |
| | <p>Print Snap Sends the currently snapped or loaded image to the printer with the current options.</p> <p>Options The options displayed in the Print Options window follow.</p> <p>Destination: Indicates the destination for the print. This can be either the printer (the default), or the name of a file. If the Printer option is specified, then the printer name can be given, or if the File option is specified, the name of the directory and file to use can be given.</p> <p>Orientation: Specifies the position of the image on the paper. The snap can be printed either up and down the page (the default), or sideways across the page.</p> <p>Position: Specifies the position of the snapshot on the page. By default the image will be centered, but you can specify the position in inches from the left and top of the paper.</p> <p>Scale to: Allows the user to specify whether the image should be printed at actual size, or if a width or height (or both) should be given, which will override the default size.</p> <p>Double Size: Specifies whether the snapshot is to be doubled in width and height.</p> <p>Monochrome Printer Select this option if the printed output is to be in monochrome.</p> |
| | <p>Snap Type: The type of image to snap; a window, a rectangular region of the screen, or the entire screen.</p> <p>Snap Delay: Delays the snapping of the picture for the specified number of seconds.</p> <p>Beep During Countdown Indicates if a beep should be sounded every second as the snap delay is counted down.</p> <p>Hide Window During Capture The snapshot window is made invisible before the snapshot is taken. The timer delay is forced to eight seconds.</p> |
| Snap | Snap the contents of the window, region or screen. If you are snapping a window, then select the desired window using SELECT . To cancel the selection, use ADJUST or MENU . If you are snapping a region, select the region to be |

snapped by using **SELECT** and dragging the rectangle around the area. Use **ADJUST** to take the snapshot or use **MENU** to cancel.

View... Launches **imagetool** and displays the snapped or loaded image in the **imagetool** main window. Once **imagetool** is running, all snapped images will be displayed in this window. Also, the **Load**, **Save** and **Print** buttons on the **snapshot** main window all become inactive once **imagetool** is running. Users can find these options on the **imagetool** file menu.

SEE ALSO **imagetool(1)**, **filemgr(1)**, **printtool(1)**

OpenWindows user documentation

"About Snapshot" in the Help Handbook available through the Help option on the Workspace menu.

| | |
|--------------------|---|
| NAME | spider – play double deck solitaire |
| SYNOPSIS | spider [-save_file <i>filename</i>] [-toolkitoption...] |
| DESCRIPTION | <p>spider is a particularly challenging double-deck solitaire. Unlike most solitaires, it provides extraordinary opportunities for the skillful player to overcome bad luck in the deal by means of careful analysis and complex manipulations. The spider program does not actually play the game for you, but rather takes the place of the cards (by displaying the tableau on the screen) and keeping a record of the game so that you can follow out long lines of play without losing track of where you started from.</p> <p>spider when compiled with XView has a property sheet for defining resources. The property sheet is on the "File" menu item. To be sure spot help is active, set the environment variable HELPDIR for XView version 1.0.1 or HELPPATH for XView version 2.0 or later to the directory containing the spider.info file. See spot help on the property sheet for more details.</p> |
| OPTIONS | <p>-save_file <i>filename</i> Start up using the specified save file.</p> |
| RESOURCES | <p>spider understands all of the core X Toolkit resource names and classes as well as:</p> <p>bell (class Bell) Boolean which specifies whether spider will use the bell to when complaining about illegal moves. The default is "true".</p> <p>confirm (class Confirm) Boolean which specifies whether spider will use ask for confirmation before certain actions. The default is "true".</p> <p>replayTime (class ReplayTime) Specifies the time (in microseconds) to pause between each move when showing a Replay. The default is 200.</p> <p>roundsCards (class RoundCards) Specifies whether to use rounded cards are not. Rounded cards look better, but are considerably slower. The default is "on".</p> <p>deltaMod (class DeltaMod) Specifies the interval at which to recalculate the inter-card spacing. The default is 1, which means the card stacks are resized every time they grow or shrink when they near the bottom of the table. Setting this to a higher value will cut down on the number of redraws by limiting visibility when its unnecessary.</p> <p>squish (class Squish) Specifies whether to use a different card layout that saves on screen space but can also be somewhat confusing. The default is "off".</p> <p>helpDir (class HelpDir) Specifies where to look for the help files.</p> |

SEE ALSO**xsol(1)****COPYRIGHT**

Copyright (c) 1990 by David Lemke & Network Computing Devices, Inc.
(lemke@ncd.com)

Copyright 1990 Heather Rose and Sun Microsystems, Inc.

Copyright (c) 1989, Donald R. Woods and Sun Microsystems, Inc.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

| | |
|---------------------|---|
| NAME | tapetool – OpenWindows tool for reading files from tape or archiving files to tape |
| SYNOPSIS | tapetool [<i>generic-tool-arguments</i>] |
| AVAILABILITY | This command is available with the OpenWindows environment. All OpenWindows standard tools use the Graphical User Interface (GUI). |
| DESCRIPTION | tapetool is an OpenWindows based XView application that uses the tar (1) command to write files to tape, extract files from tape or list the contents of a tape. |
| OPTIONS | <i>generic-tool-arguments</i> tapetool accepts the generic tool arguments described in xview (7). |
| USAGE | <p>tapetool is part of the OpenWindows user environment.</p> <p>There are two windows associated with tapetool. The main window that appears when the tool is initially brought up is for writing files to tape. The read window appears when you 'List' the contents of a tape or choose either 'Read Selected' or 'Read Entire List'.</p> <p>'List...' panel-button brings up a 'Tape Contents' window that lists the files on tape. Filenames can be added using the 'File To Read' button. Filenames can be deleted using the scrolling list popup menu. It may take some time listing the entire contents of a tape if it contains many files. If you want to extract just a few files and know the exact names of the files you may want to do a 'Read Selected' that will bring up an empty 'Tape Contents' window. Then add the filenames to the list using the 'File To Read' button and do a 'Read Entire List'.</p> <p>There are three ways to read files:</p> <p style="padding-left: 20px;">Read Selected Reads the files selected in the 'Tape Contents' window.</p> <p style="padding-left: 20px;">Read Entire List Reads all the files listed in the 'Tape Contents' window.</p> <p style="padding-left: 20px;">Read Entire Tape Reads the entire tape. The 'Tape Contents' window does not have to be showing to do this. The files are put into the directory specified by the 'Destination' field if they do not contain fully qualified path-names.</p> <p>To write files to tape, add filenames to the main scrolling list using the 'File To Write' menu button. The main scrolling list has a popup menu for deleting selected files. 'Write' writes the filenames listed in this window to tape.</p> <p>Props gives a property sheet for specifying tar(1) options.</p> <p style="padding-left: 20px;">Device: /dev/rmt/<i>n</i> SCSI tape drives</p> <p style="padding-left: 20px;">Host Name: Specifies the name of host system where the reading or archiving of</p> |

files will be applied. The default is the current host name.

Tar Options

Write:

- | | |
|------------------|--|
| No SCCS | F option to tar(1) , exclude all directories named SCCS from tarfile. |
| No SCCS+ | FF option to tar(1) , exclude all directories named SCCS, all files with .o as their suffix, and all files named errs,core , and a.out . |
| Block I/O | b option to tar(1) , blocking factor for tape records. The default blocking factor is 20 blocks. The block size is determined automatically when reading tapes. This determination of the blocking factor may be fooled when reading from a pipe or a socket. The maximum blocking factor is determined only by the amount of memory available to tar when it is run. Larger blocking factors result in better throughput, longer blocks on nine-track tapes, and better media utilization. |
| Sym Links | h option to tar(1) , follow symbolic links as if they were normal files or directories. Normally, tar does not follow symbolic links. |
| Show Errs | l option to tar(1) , Display error messages if all links to archived files cannot be resolved. If not used, no error messages are printed. |
| Suppress | o option to tar(1) , Suppress information specifying owner and modes of directories that tar normally places in the archive. |

Read:

- | | |
|--------------------------|--|
| No Check Mod Time | i option to tar(1) , Ignore directory checksum errors. |
| Orig Mode | m option to tar(1) , Do not extract modification times of extracted files. The modification time will be the time of extraction. |
| | p option to tar(1) , Restore the named files to their original modes, ignoring the present umask(2) . SetUID and sticky information are also extracted if you are the super-user. |

Delete Dir:

- | | |
|------------------|---|
| | For writing files to tape |
| 'NONE' | retains filename path exactly as specified. |
| 'ALL' | Extracts the entire path from <i>filename</i> and writes just the <i>filename</i> . |
| 'PATTERN' | extracts the <i>pattern</i> specified and writes the <i>filename</i> left over. |

Other:

- | | |
|-----------------|---|
| Err Exit | e option to tar(1) , If any unexpected errors |
|-----------------|---|

occur **tar** aborts immediately with a positive exit status.

Exclude

X option to **tar(1)**, Specify a file containing a list of named files (or directories) to be excluded when reading from tape. Only one file can be specified.

Apply Menu button that applies the options that have been set.

Reset Menu button that resets the defaults.

You may drag and drop files from **filemgr(1)** onto the tapetool write window for writing files to tape. You can also drop files onto the **tapetool** icon for writing.

FILES

/dev/rmt/n SCSI tape interface

BUGS

See bugs for **tar(1)**

SEE ALSO

tar(1), **filemgr(1)**

OpenWindows user documentation

"About Tape Tool" in the Help Handbook available through the Help option on the Workspace menu.

| | |
|-------------|---|
| NAME | textedit – XView-based text editor with mouse support |
| SYNOPSIS | <pre>textedit [<i>generic-tool-arguments</i>] [-Ei on off] [-auto_indent] [-Eo on off] [-okay_to_overwrite] [-Er on off] [-read_only] [-Ec N] [-checkpoint count] [-EL lines] [-lower_context lines] [-Em pixels] [-margin pixels] [-En N] [-number_of_lines lines] [-ES N] [-multi_click_space radius] [-Et N] [-tab_width tabstop] [-ET N] [-multi_click_timeout intrvl] [-Eu N] [-history_limit max] [-EU N] [-upper_context lines] <i>filename</i></pre> |
| DESCRIPTION | <p>textedit is a window-based XView application that provides both mouse and pointer operations and keyboard accelerators for the editing of text. In general, text editing throughout the OpenWindows user environment uses textedit conventions, both in text display areas such as mail message windows and command panel text fields.</p> <p>textedit operates via a set of command panel buttons and text fields and a set of menus and submenus accessible from the main editing window.</p> |
| OPTIONS | <p><i>generic-tool-arguments</i></p> <p>textedit accepts the XView generic tool arguments described in the xview(7) man page with the exception of the arguments for setting the frame label.</p> <p>-Ei on off</p> <p>-auto_indent Choose whether or not to automatically indent newly-opened lines. The default is off. Corresponds to the auto_indent Text default.</p> <p>-Eo on off</p> <p>-okay_to_overwrite Set behavior to the Store as New File menu item. If on a Store as New File to the current file is treated as a Save Current File. If off (the standard default), Store as New File operations using the current filename results in an error message. Corresponds to Store_self_is_save.</p> <p>-Er on off</p> <p>-read_only Turn read-only mode on or off. When on, text cannot be modified.</p> <p>-Ec N</p> <p>-checkpoint count Checkpoint after every <i>count</i> editing operation. If <i>count</i> is 0 (the standard default), no checkpointing takes place. Each character typed, each Paste, and each Cut counts as an editing operation. Corresponds to checkpoint_frequency.</p> <p>-EL lines</p> <p>-lower_context lines Specify the minimum number of lines to keep between the caret and the bottom of the text subwindow. The default is 2. Corresponds to lower_context.</p> |

-Em pixels
-margin pixels
 Set the scrollbar margin width in pixels. The default is 4. Corresponds to **left_margin**.

-En N
-number_of_lines lines
 Set the number of lines in the bottom subwindow. The default is 45.

-ES N
-multi_click_space radius
 Set the radius in pixels, within which clicks must occur to be treated as a multi-click selection. The default is 3 pixels. Corresponds to **multi_click_space**.

-Et N
-tab_width tabstop
 Set the number of SPACE characters displayed per TAB stop. The default is 8. This option has no effect on the characters in the file. Corresponds to **tab_width**.

-ET N
-multi_click_timeout intrvl
 Set the interval, in milliseconds, within which any two clicks must occur to be treated as a multi-click selection. The default is 390 milliseconds. Corresponds to **multi_click_timeout**.

-Eu N
-history_limit max
 Set the maximum number of editing operations that can be undone or replayed. The default is 50. Corresponds to **history_limit**.

-EU N
-upper_context lines
 Set the minimum number of lines to keep between the caret and the top of the text subwindow. The default is 2. Corresponds to **upper_context**.

USAGE **textedit** is part of the OpenWindows user environment.

Signal Processing If **textedit** hangs, for whatever reason, you can send a **SIGHUP** signal to its process ID, which forces it to write any changes (if possible):

kill -HUP pid

The edits are written to the file **textedit.pid** in its working directory. If that fails, **textedit** successively tries to write to a file by that name in **/var/tmp**, and then **/tmp**. In addition, whenever **textedit** catches a fatal signal, such as **SIGILL**, it tries to write out the edits before aborting.

Defaults Options There are several dozen user-specified defaults that affect the behavior of the text-based facilities. See **xview(7)** for a complete description. Important defaults entries in the **Text** category are:

Checkpoint_frequency

If set to **0** (the standard default) no checkpointing is done. For any value greater than zero, a checkpoint is made each time the indicated number of editing operations has been performed since the last checkpoint. Each character typed, each **Paste**, and each **Cut** counts as an editing operation. The checkpoint file has a name of the form: *filename%%*, where *filename* is the name of the file being edited.

Selections

Selections in **textedit** are defined as selected portions of text to which editing operations can be applied. For example, a selection can be *deleted, moved, copied*, etc.

textedit provides two types of selections: primary and secondary. Primary selections allow you to select a set of text on which to perform an editing function. Secondary selections allow you to define a *second* block of text without undefining your primary text selection or repositioning your cursor. Being able to define two sets of text at once allows you to take advantage of the advanced editing functions described below in the section called **Function Keys**.

Using a Mouse and Pointer:

Single characters can be selected using the SELECT mouse button.

Blocks of text can be selected by selecting a starting point with the SELECT mouse button and selecting an ending point with the ADJUST button.

Or blocks of text can be selected using OPEN LOOK's *wipe through* feature by pointing at a beginning character and depressing the SELECT button while moving the pointer to the end of a block of text.

Selections can also be made by *clicking* (rapidly pressing) the SELECT button. Click once to select a single letter; click twice to select a word; click three times to select a complete line of text; click four times to select the entire document being edited.

Visual Feedback

All primary selections are indicated visually by inverse video of the text selected and are *pending delete*. Pending delete selections are replaced if you type or paste while the text is selected.

Secondary selections that are not *pending delete* are indicated by underlining of the text.

Secondary selections pending delete are indicated by underlining of the text and strike through of the individual characters.

Secondary Selections

Secondary selections are made using any of the selection methods described above in addition to holding down one of the four function keys corresponding to the commands **Cut**, **Find**, **Paste**, or **Copy**.

Secondary selections are made pending delete by holding the CTRL key when making the secondary selection. If a secondary selection is pending-delete, it is deleted when the function key is released, except in the case of the **Find**, which deselects the secondary selection.

**Inserting Text and
Command Characters**

Commands that operate on the primary selection do so even if the primary selection is not in the window that issued the command.

For the most part, typing any of the standard keys either inserts the corresponding character at the insertion point, or erases characters. However, certain key combinations are treated as commands. Some of the most useful are:

| Command | Character | Description |
|--------------------------|------------------|---|
| Cut-Primary | META-X | Erases the primary selection and moves it to the Clipboard. |
| Find-Primary | META-F | Searches the text for the pattern specified by the primary selection or by the Clipboard, if there is no primary selection. |
| Copy-to-Clipboard | META-C | Copies the primary selection to the Clipboard. |
| Paste-Clipboard | META-V | Inserts the Clipboard contents at the insertion point. |
| Copy-then-Paste | META-P | Copies the primary selection to the insertion point (through the Clipboard). |
| Go-to-EOF | CTRL-RETURN | Moves the insertion point to the end of the text and positions the text so the insertion point is visible. |

Keyboard Functions

The commands indicated by use of the function keys are:

| Command | SPARC | x86 | Description |
|----------------|--------------|------------|--|
| Stop | L1 | Esc | Aborts the current command. |
| Again | L2 | Meta-a | Repeats the previous editing sequence since a primary selection was made. |
| Undo | L4 | Meta-z | Undoes a prior editing sequence. |
| Front | L5 | Meta-r | Makes the window completely visible (or hides it, if it is already exposed). |
| Copy | L6 | Meta-c | Copies the primary selection, either to the Clipboard or at the closest end of the secondary selection. |
| Open | L7 | Meta-w | Makes the window iconic (or normal, if it is already iconic). |
| Paste | L8 | Meta-v | Copies either the secondary selection or the Clipboard at the insertion point. |
| Find | L9 | Meta-f | Searches for the pattern specified by, in order, the secondary selection, the primary selection, or the Clipboard. |
| Cut | L10 | Meta-x | Erases either the primary or the secondary selection and moves it to the Clipboard. |
| Help | F1 | F1 | Produces help text. |

Find usually searches the text forwards, towards the end. Holding down the SHIFT key while invoking **Find** searches backward through the text, towards the beginning. If the pattern is not found before the search encounters either extreme, it “wraps around” and continues from the other extreme. **Find** starts the search at the appropriate end of the

primary selection, if the primary selection is in the subwindow that the search is made in; otherwise it starts at the insertion point, unless the subwindow cannot be edited, in which case it starts at the beginning of the text.

CTRL-Find invokes the **Find and Replace** pop-up frame.

Menu Items

- File** A pull-right menu item for file operations.
- Edit** A pull-right menu item equivalent of the editing function keys. The **Edit** submenu provides **Again**, **Undo**, **Copy**, **Paste**, and **Cut** (same as function keys L2, L4, L6, L8, and L10).
- Display** A pull-right menu item for controlling the way text is displayed and line display format.
- Find** A pull-right menu item for find and delimiter matching operations.
- Extras** A user definable pull-right menu item. The **Extras** standard submenu is controlled by `$OPENWINHOME/lib/locale/<locale>/xview/.text_extras_menu`. This file has the same syntax as `.openwin-menu` file. See the `xview` man page.

Only those items that are active appear as normal text in the menu; inactive items (which are inappropriate at the time) are "grayed out".

User Defined Commands

The file `$OPENWINHOME/lib/locale/<locale>/xview/.text_extras_menu` specifies filter programs that are included in the text subwindow **Extras** pull-right menu item. The file `~/textswrc` specifies filter programs that are assigned to (available) function keys. These filters are applied to the contents of the primary selection. Their output is entered at the caret.

The file `$OPENWINHOME/share/src/xview/examples/textsw/textswrc` is a sample containing a set of useful filters. It is not read automatically.

FILES

- `~/textswrc` specifies bindings of filters to function keys
- `$OPENWINHOME/lib/locale/<locale>/xview/.text_extras_menu` specifies bindings of filters for the extras menu pull-right items
- `$OPENWINHOME/share/src/xview/examples/textsw/textswrc` contains useful filters, including **shift_lines** and **capitalize**.
- `filename%` prior version of `filename` is available here after a **Save Current File** menu operation
- `textedit.pid` edited version of `filename`; generated in response to fatal internal errors
- `/tmp/Text*` editing session logs

SEE ALSO

kill(1), **xview(7)**
Solaris User's Guide
 "About Text Editor" in the Help Handbook available through the Help option on the Workspace menu.

DIAGNOSTICS

Cannot open file '*filename*', aborting! *filename* does not exist or cannot be read.

textedit produces the following exit status codes:

- 0 normal termination
- 1 standard OpenWindows help message was printed
- 2 help message was requested and printed
- 3 abnormal termination in response to a signal, usually due to an internal error
- 4 abnormal termination during initialization, usually due to a missing file or running out of swap space

BUGS

Multi-click to change the current selection does not work for **Adjust Selection**.

Handling of long lines is incorrect in certain scrolling situations.

There is no way to replay any editing sequence except the most recent.

| | |
|--------------------|---|
| NAME | textedit_filters, align_equals, capitalize, insert_brackets, remove_brackets, shift_lines – filters provided with textedit(1) |
| SYNOPSIS | <p>align_equals</p> <p>capitalize [-u -l -c]</p> <p>insert_brackets <i>l r</i></p> <p>remove_brackets <i>l r</i></p> <p>shift_lines [-t] <i>n</i></p> |
| DESCRIPTION | <p>Each of these filters is designed to operate on pending delete selections in text subwindows. You can use them from within text subwindows either by mapping them to function keys in your <code>.textswrc</code> file or adding them to the text 'Extras' menu in your <code>.text_extras_menu</code> file. When a filter is used as a command (perhaps in a pipeline), it is applied to the standard input and the filtered text appears on standard output.</p> <p>align_equals lines up the '=' (equal signs) in C assignment statements. Some programmers feel that this makes for improved readability. It aligns all equal signs with the rightmost equal sign in the selection (or the standard input), by padding with spaces between the sign and the previous nonwhite character; it replaces the selection with the aligned text (or writes this text to the standard output). For instance:</p> <pre> big_long_expression = x; shorter_expr = y; z += 1; </pre> <p>becomes:</p> <pre> big_long_expression = x; shorter_expr = y; z += 1; </pre> <p>capitalize changes the capitalization of the selection (or the standard input) and replaces it (or writes to the standard output). The -l option converts all characters to lower case; -c converts the first letter of each word to upper case; and -u converts all characters to upper case. If no option is specified, then capitalize consults its input to determine what to do. If the text is all capitals, it is converted to all lower case. If the text is all lower case or of mixed cases and contains no white space (such as a NEWLINE, SPACE, or TAB), it is converted to all capitals. If there is white space, then the case of the first character in each word is inverted.</p> <p>insert_brackets surrounds the selection (or the standard input) with the specified character sequences. <i>l</i> and <i>r</i> are the left- and right-bracketing characters, respectively.</p> <p>remove_brackets removes the left- and right-bracketing characters, specified by <i>l</i> and <i>r</i>, respectively from the selection (or the standard input).</p> <p>shift_lines adjusts indentation of the selection (or the standard input) by <i>n</i> spaces, and replaces the selection with the adjusted text (or writes to the standard output).</p> <p>shift_lines adjusts to the left when <i>n</i> is negative. If -t is specified, the lines are shifted left or right by <i>n</i> tab stops. The default is 8 spaces per tab stop, but if the first line of the</p> |

selection (or the standard input) begins with white space, then the tab stops are set to four spaces.

FILES /tmp/Cap.pid temporary file used by **capitalize**
 /tmp/Ins.pid temporary file used by **insert_brackets**
 \$OPENWINHOME/lib/locale/<locale>/xview/.text_extras_menu
 default 'Extras' menu
 \$OPENWINHOME/share/src/xview/examples/textsw/textswrc
 sample function-key mappings

SEE ALSO textedit(1)

| | |
|--------------------|---|
| NAME | texteroids – test your mousing skills on spinning text |
| SYNOPSIS | texteroids [-display <i>name</i>] [-fn <i>font</i>] [-size <i>size</i>] [<i>text_string</i>] |
| DESCRIPTION | texteroids spins the specified text string in a window. If you click on the text with the mouse, the string splits up into individual letters, each of which you may then click on. |
| OPTIONS | -display <i>name</i> specifies the display on which to open a connection to the Display PostScript System. If no display is specified, the DISPLAY environment variable is used. -fn <i>font</i> specifies the name of the PostScript language font software to use. The default is Times-Italic. -size <i>size</i> specifies the size, in points, of the text. The default is 36. <i>text_string</i> specifies the text to display. If the text has spaces it must be enclosed in quotation marks. The default text string is "Adobe". |
| AUTHOR | Adobe Systems Incorporated |
| NOTES | PostScript and Display PostScript are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions. Copyright (c) 1990-1991 Adobe Systems Incorporated. All rights reserved. |

| | |
|--------------------|--|
| NAME | toolwait – control client program startup |
| SYNOPSIS | toolwait [-display <i>displaystring</i>] [-timeout <i>nseconds</i>] [-help] <i>command</i> |
| DESCRIPTION | toolwait is a program that controls the startup of an X11 client program. toolwait takes a client program as its argument. When the client program has started, toolwait will exit. |
| OPTIONS | The basic options are as follows: -display <i>string</i> Specify the name of the display that toolwait should monitor. Overrides the DISPLAY environment variable, if any. -timeout <i>nsecs</i> Specifies the time to wait for the client to start up. Toolwait will exit if the client hasn't started by <i>nsecs</i> seconds. Default is 15 seconds. -help Display a short summary of the possible command line options. |
| EXAMPLES | To start a cmdtool with toolwait , the following command may be used: <pre>example% toolwait cmdtool -Wp 0 0 -Ws 557 95 -WP 263 833</pre> To start a mailtool on display "host2" with a timeout of 25 seconds, use the following command: <pre>example% toolwait -display host2:0 -timeout 25 mailtool -display host2:0</pre> |
| DIAGNOSTICS | Exit status is 0 for clean exits, the exit status of the child if the child has problems, and 1 if toolwait has problems. |
| NOTES | In the current version of toolwait , a client is considered "started" when it has mapped a top level window on the display. As soon as toolwait detects that a top level window has been mapped, it will exit. |

| | |
|--------------------|--|
| NAME | ttce2xdr – convert ToolTalk Classing Engine type tables to XDR format |
| SYNOPSIS | <pre>ttce2xdr [-xn] -d user system ttce2xdr [-xn] -d network [OPENWINHOME-from [OPENWINHOME-to]] ttce2xdr [-h] ttce2xdr [-v]</pre> |
| DESCRIPTION | <p>ttce2xdr converts ToolTalk types stored in the Classing Engine data base, used by ToolTalk in versions 1.0.x, to the XDR-format data base used in version 1.1 and later. For user type data bases, this conversion is done automatically the first time a version 1.1 tsession is started for the user, so this command generally only needs to be used by administrators updating types data bases common to systems or the network.</p> <p>The first format is used to convert user or system data bases for the current user or current system. The second format is used to convert the network-wide data base, and provides additional options allowing types to be taken from one data base and stored into another.</p> |
| OPTIONS | <pre>-x Show the underlying commands executed by ttce2xdr. -v Print version and exit. -n Just show underlying commands that would be executed by ttce2xdr. -h Describe the options for ttce2xdr and exit. -d Specify the data base to work on which should be one of user (default), system, or network. The types are read from the Classing Engine data base, which is: user -- ~/.cetables/cetables system -- /etc/cetables/cetables network -- \$OPENWINHOME/lib/cetables/cetables</pre> <p>The types are written to the XDR data base which is:</p> <pre>user -- ~/.tt/types.xdr system -- /etc/tt/types.xdr network -- \$OPENWINHOME/etc/tt/types.xdr</pre> <p>If the network data base is specified, the optional arguments @and @may be specified. If neither is specified, the current value of the environment variable OPENWINHOME is used to locate the data bases to be read and written. If only OPENWINHOME-from is specified, the data bases under the directory named by OPENWINHOME-from are read and written. If both are specified, the data base under OPENWINHOME-from is read and the data base under OPENWINHOME-to is written.</p> |

ENVIRONMENT**CEPATH**

In Classing Engine mode, **tt_type_comp** will use this variable for its definition of where the databases are located. See **ce_db_build(1)**. **OPENWINHOME** Location of network data bases.

FILES

~/tt/types.xdr User's ToolTalk XDR format types file
/etc/tt/types.xdr System ToolTalk XDR format types file
\$OPENWINHOME/etc/tt/types.xdr
"Network-wide" ToolTalk XDR format types file
\$OPENWINHOME/lib/cetables/cetables
Classing Engine database containing the ToolTalk type definitions

SEE ALSO

tt_type_comp(1), **ttsession(1)**, **ce_db_build(1)**, **ce_db_merge(1)**

| | |
|--------------------|--|
| NAME | ttcp – copy files in a ToolTalk-safe way |
| SYNOPSIS | ttcp [-pL] <i>filename1 filename2</i> ttcp -r [-pL] <i>directory1 directory2</i> ttcp [-prL] <i>filename ... directory</i> ttcp -v ttcp -h |
| DESCRIPTION | ttcp invokes cp(1) to copy files and directories, and informs ToolTalk about its actions so that the TT objects associated with the named files and directories can be duplicated at the same time. |
| OPTIONS | <p>-p Preserve. Duplicate not only the contents of the original file or directory, but also the modification time and permission modes. (The modification times of TT objects are preserved only if you are superuser.)</p> <p>-r Recursive. If any of the source files are directories, copy the directory along with its files (including any subdirectories and their files); the destination must be a directory.</p> <p>-L Copy the TT objects of the files, but do not invoke cp(1) to copy the actual files.</p> <p>-v Print the version of ttcp.</p> <p>-h Print usage information for ttcp.</p> |
| FILES | /mountpoint/TT_DB This directory is used as a database for the ToolTalk objects of files in the filesystem mounted at <i>/mountpoint</i> . |
| SEE ALSO | cp(1) , ttmv(1) , tttar(1) , ttsession(1) |
| BUGS | <p>The -i (interactive) option of cp(1) is not supported.</p> <p>The -p option does not preserve the modification times of TT objects unless you are superuser.</p> |

| | |
|--------------------------|---|
| NAME | ttdbck – display, check, or repair ToolTalk databases |
| SYNOPSIS | ttdbck [<i>selection opts</i>] [<i>diagnosis opts</i>] [<i>display opts</i>] [<i>repair opts</i>] [<i>data-base-directory</i>]... |
| DESCRIPTION | ttdbck is the ToolTalk database maintenance tool. It allows direct inspection of ToolTalk spec data, detection of inconsistencies, and repair of problems. |
| OPTIONS | <p><i>data-base-directory</i></p> <p>Names the directory or directories containing the ToolTalk database to be inspected or repaired. If no directories are named, the current directory is assumed. If a directory path does not end in “TT_DB”, “TT_DB” is appended.</p> <p>The user running the command must have read access to the files in the directory to inspect the data and write access to repair the data. Since ToolTalk databases are typically accessible only to root, this command is normally run as root.</p> |
| Selection options | <p>The selection options determine which specs in the database are displayed or modified. If no selection options are given, all specs in the database are displayed. To prevent massive accidental changes to ToolTalk databases, no repair options except -I are allowed unless a selection or diagnosis option is given.</p> <p>-f filename</p> <p>Restricts the set of specs to be inspected or modified to those which describe objects in the named file. The file name can contain shell-style wildcards which must be escaped to prevent the shell from expanding them.</p> <p>-k objidkey</p> <p>An object id key, specifying a particular spec to be displayed or modified. The object id key can be obtained from a previous invocation of ttdbck; one might display a set of specs, determine the one that needs repair, and specify its key here.</p> <p>-t type</p> <p>Restricts the set of specs to be inspected or modified to those with otype <i>type</i>. The type name can contain shell-style wildcards which must be escaped to prevent the shell from expanding them.</p> |
| Diagnosis options | <p>These options check for and report on inconsistencies in the selected specs. Only specs selected by the selection options are checked. If a diagnosis option is given, any display or repair option is applied only to specs which fail the diagnostic check.</p> <p>-b</p> <p>Check for badly formed specs: those which have no file or type or those which have types not defined in the type database.</p> <p>-x</p> <p>Check for specs which refer to files that no longer exist.</p> |
| Display options | <p>These options determine which data is printed for each selected spec.</p> <p>-i</p> <p>Display the object id (including the object id key.)</p> <p>-m</p> <p>Display the mandatory data that must appear in every spec: the otype of the object described by the spec and the file in which the spec is stored.</p> <p>-p</p> <p>Display all the properties and values for each selected spec.</p> <p>-a</p> <p>Display all data (equivalent to specifying -imp)</p> |

Repair options

- I** Invoke the NetISAM `isrepair()` function for all files accessed. This action is applied before any other inspection or repair action. This option should be used when normal operations return EBADFILE (error code 105).
- F *filename***
Change the file name for the selected specs to the supplied file name.
- T *otypeid***
Change the type of the selected specs to the given otype.
- Z** Remove the selected specs entirely.

EXAMPLES

ttdbck -bxi /home

In the `/home/TT_DB` directory, finds all badly formed specs and specs that refer to non-existent files and prints their ids.

ttdbck -f /home/sample/data -F /home/sample/data1 /home

In the `/home/TT_DB` directory, finds all specs that refer to objects in file `/home/sample/data` and changes them to refer to `/home/sample/data1`.

ttdbck -t Sun_Chain_Link -Z /export/TT_DB

In the `/export/TT_DB` directory, finds all specs that refer to objects of type `Sun_Chain_Link` and deletes the specs.

FILES

`/path/TT_DB` ToolTalk database

NOTES

The **ttdbck** command should be run on the same machine where the `TT_DB` files being inspected and repaired physically exist. That is, don't try to access the `TT_DB` files via NFS.

| | |
|--------------------|--|
| NAME | ttdbserverd, rpc.ttdbserverd – RPC-based ToolTalk database server |
| SYNOPSIS | rpc.ttdbserverd [-n] [-t level] [-?] [-v] |
| DESCRIPTION | rpc.ttdbserverd is the ToolTalk database server daemon. This process (normally started by inetd) performs all database operations for databases stored on the host. |
| OPTIONS | -n Turn off permission checking. Normally the protection of the file containing the spec determines who may read and write that spec. This option disables this checking and allows anyone to read and write any spec. This option should be used with caution. -t level Turns on trace level, the valid level is between 1 to 10. -? Prints out the command usage information. |
| FILES | TT_DB/* The NetISAM database files are kept in the TT_DB directory under each disk partition mount point. |
| SEE ALSO | ttsession(1) |

| | |
|--------------------|--|
| NAME | ttmv – move or rename files in a ToolTalk-safe way |
| SYNOPSIS | ttmv [-] [-fL] <i>pathname1 pathname2</i> ttmv [-] [-fL] <i>pathname ... directory</i> ttmv -v ttmv -h |
| DESCRIPTION | ttmv invokes mv (1) to move files and directories around in the file system and informs ToolTalk about its actions so that the TT objects associated with the itinerant files and directories can be moved at the same time. |
| OPTIONS | - Treat the following arguments as filenames so that you can specify filenames starting with a minus. -f Force. Do not report any errors, and pass the -f option to mv (1). -L Move the TT objects of the files, but do not invoke mv (1) to move the actual files. -v Print the version of ttmv . -h Print usage information for ttmv . |
| FILES | /mountpoint/TT_DB This directory is used as a database for the ToolTalk objects of files in the filesystem mounted at <i>/mountpoint</i> . |
| SEE ALSO | mv (1), ttsession (1) |
| BUGS | The -i (interactive) option of mv (1) is not supported. ttmv moves the TT objects before it moves the files and does not check whether the latter operation will succeed before performing the former. |

| | |
|--------------------|--|
| NAME | ttrm, ttrmdir – remove files or directories in a ToolTalk-safe way |
| SYNOPSIS | ttrm [-] [-frL] <i>pathname</i> ... rmdir <i>directory</i> ... ttrm [<i>dir</i>] -v ttrm [<i>dir</i>] -h |
| DESCRIPTION | ttrm invokes rm (1) or rmdir (1) to remove files and directories and informs ToolTalk about its actions so that the TT objects associated with the deleted files and directories can be deleted at the same time. |
| OPTIONS | - Treat the following arguments as filenames so that you can specify filenames starting with a minus. -f Force. Do not report any errors, and pass the -f option to rm (1). -r Recursively delete the TT objects of any directories named, and pass the -r option to rm (1). -L Remove the TT objects of the files or directories, but do not invoke rm (1) or rmdir (1) to remove the actual files or directories. -v Print the version of the command. -h Print usage information for the command. |
| FILES | /mountpoint/TT_DB This directory is used as a database for the ToolTalk objects of files in the filesystem mounted at <i>/mountpoint</i> . |
| SEE ALSO | rm (1), rmdir (1), ttsession (1) |
| BUGS | The -i (interactive) option of rm (1) is not supported. ttrm and ttrmdir remove the TT objects before they remove the files and do minimal checking of whether the latter operation will succeed before performing the former. |

| | |
|--------------------|--|
| NAME | ttsample1 – simple ToolTalk demo |
| SYNOPSIS | ttsample1 |
| DESCRIPTION | ttsample1 is a program provided as demo code for the ToolTalk product. This program is compiled by running the make(1S) command in the directory <code>\$OPENWINHOME/share/src/tooltalk/demo/ttsample1</code> . Your <code>OPENWINHOME</code> environment variable must be set to where OpenWindows is installed (typically <code>/opt/openwin</code> or <code>/usr/openwin</code>). After compiling, run the ttsample1 program. This will popup an OpenWindows application with a single button, slider, and a scale reflecting how many messages were sent and received. You can use ttsnoop(6) to monitor the contents of the ToolTalk messages sent by ttsample1 . |
| SEE ALSO | ttsnoop(6) ttsession(1) |
| DIAGNOSTICS | If you try and invoke ttsample1 (or any ToolTalk application) and you get a message saying the application could not start ToolTalk, or ttsession , make sure that you have one of the environment variables <code>DISPLAY</code> or <code>_SUN_TT_SESSION</code> set, and that ttsession is in your <code>PATH</code> , or that the <code>SUN_TTSESSION_CMD</code> environment variable indicates where the ttsession program resides. For more information on ttsession and the environment variables it uses, see the ttsession man page. |

| | |
|--------------------|---|
| NAME | ttsession – the ToolTalk message server |
| SYNOPSIS | ttsession [<i>-a level</i>] [<i>-d display</i>] [<i>-spStvhN</i>] [- { E X }] [<i>-c [command]</i>] |
| DESCRIPTION | ttsession is the ToolTalk message server. This background process must be running before any messages can be sent or received. Each message server defines a "session". The message server has no user interface and typically runs in the background, started either by the user's .xinitrc file or automatically by any program which needs to send a message. The message server reacts to two signals. If it receives the <i>USR1</i> signal, it toggles trace mode on or off (see the <i>-t</i> option below). If it receives the <i>USR2</i> signal, it rereads the types file. |
| OPTIONS | <p><i>-a level</i> Set the server authentication level. The level must be <i>unix</i>, <i>xauth</i>, or <i>des</i>.</p> <p><i>-c</i> [<i>command</i>] Starts a process tree session and runs the given command. The special environment variable <code>_SUN_TT_SESSION</code> will be set to the name of this session. Any process started with this variable in the environment will default to being in this session. If <i>command</i> is omitted the value of <code>\$SHELL</code> is used instead. Everything after <i>-c</i> on the command line is taken as the command to be executed, so <i>-c</i> should be the last option.</p> <p><i>-p</i> Prints the name of a new process tree session on stdout, and then forks a background instance of ttsession to manage this new session.</p> <p><i>-S</i> Directs ttsession not to fork a background instance to manage its session.</p> <p><i>-N</i> Maximizes the number of clients allowed to connect to (i.e. open procds in) this session, by raising the soft limit on the number of open file descriptors (<code>RLIMIT_NOFILE</code>) all the way to the hard limit. See <code>setrlimit(2)</code>. On SunOS 5.x, the increase is from about 50 to about 500. The default is not to maximize, because some of the system libraries allocate up front an amount of storage that is linear in <code>RLIMIT_NOFILE</code>, which hurts the paging performance of ttsession.</p> <p><i>-d display</i> Directs ttsession to start an X session for the given display. Normally, ttsession uses the <code>DISPLAY</code> environment variable.</p> <p><i>-s</i> Silent - don't print any warning messages.</p> <p><i>-t</i> Turn on trace mode. Tracing is very helpful for seeing how messages are dispatched and delivered. The output is very voluminous. See the signals above for how to turn tracing on and off during execution. Tracing displays the state of a message when it is first seen by ttsession. The lifetime of the message is then shown by showing the result of matching the message against type signatures (dispatch stage) and then showing the result of matching the message against any registered message patterns (delivery stage). Any attempt to send the message to a given process is also shown together with the success of that attempt.</p> <p><i>-X</i> Read in the types from the XDR format databases in <code>\$HOME/.tt/types.xdr</code>,</p> |

/etc/tt/types.xdr, and \$OPENWINHOME/etc/tt/types.xdr. These three locations can be overridden by setting \$TTPATH. See Environment, below.

- E** Read in the types from the Classing Engine database. If neither **-E** nor **-X** is given, **-X** is assumed.
- v** Print out the version number and exit.
- h** Print help on invoking **ttsession** and exit.

ENVIRONMENT

SUN_TTSESSION_CMD

If set, this will be used by all ToolTalk clients as the command to use for auto-starting **ttsession**.

TTPATH

In XDR mode, a colon-separated list of directories that tells ToolTalk where to find the ToolTalk types databases. The format of this variable is userDir[:systemDir[:networkDir]].

CEPATH

In Classing Engine mode, this variable tells the Classing Engine where to find the databases that contain (among other things) ToolTalk types. See **ce_db_build(1)**.

_SUN_TT_ARG_TRACE_WIDTH

How many characters of argument and context values to print when in trace mode. The default is to print the first 40 characters.

_SUN_TT_SESSION

ttsession uses this variable to communicate its session ID to the tools that it starts. If this variable is set, the ToolTalk client library uses its value as the default session ID.

DISPLAY

If **_SUN_TT_SESSION** is not set and **\$DISPLAY** is set, then **\$DISPLAY** names the X display that **ttsession** will connect to when it is auto-started by a ToolTalk client. Typically **_SUN_TT_SESSION** is set by **ttsession** when running in process tree mode, although it can be set manually by a sophisticated user, and **\$DISPLAY** is used when **ttsession** is auto-started while running under OpenWindows.

If **ttsession** is run with the **-d** option and **\$DISPLAY** is not set, **ttsession** sets **\$DISPLAY** to be the value of the **-d** option. This helps ToolTalk clients to find the right X display when they are auto-started by **ttsession**.

_SUN_TT_TOKEN

This variable lets the ToolTalk client library know it has been invoked by **ttsession**, so that the client can confirm to **ttsession** that it started successfully.

_SUN_TT_FILE

When **ttsession** invokes a tool to receive a message, it copies the file attribute (if any) of the message into this variable.

FILES

\$HOME/.tt/types.xdr User's ToolTalk types database for XDR mode
/etc/tt/types.xdr System ToolTalk types database for XDR mode
\$OPENWINHOME/etc/tt/types.xdr
 Network ToolTalk types database for XDR mode

\$HOME/.cetables/cetables
/etc/cetables/cetables
\$OPENWINHOME/lib/cetables/cetables
 Classing Engine databases containing ToolTalk types for CE mode.
 See **ce_db_build(1)**.

SEE ALSO **tt_type_comp(1)**

DIAGNOSTICS When the **-c** child process exits, **ttsession** will exit with the exit code of the dead child. Otherwise, exit codes are as follows:

- 0** Normal termination. Without the **-c** or **-S** options, a zero exit status means **ttsession** has successfully forked an instance of itself that has begun serving the session.
- 1** Abnormal termination. **ttsession** was given invalid command line options, was interrupted by SIGINT, or encountered some internal error.
- 2** Collision. Another **ttsession** was found to already be serving the session.

| | |
|--------------------|--|
| NAME | ttsnoop – the ToolTalk message monitor demo |
| SYNOPSIS | ttsnoop [-t] |
| DESCRIPTION | <p>ttsnoop is a demo program/tool provided with the ToolTalk product, which allows you to create and send custom constructed ToolTalk messages, and to selectively monitor any or all ToolTalk messages on your system.</p> <p>The ttsnoop program is compiled by running the make command in the directory <code>\$OPENWINHOME/share/src/tooltalk/demo/ttsnoop</code>. Your <code>OPENWINHOME</code> environment variable must be set to where OpenWindows is installed (typically <code>/opt/openwin</code> or <code>/usr/openwin</code>). The OpenWindows portion of ttsnoop was constructed using the Sun DevGuide product. The .G files provided with the demo are provided for the benefit of advanced users who may wish to further modify ttsnoop, and are not used in the course of normal compilation of ttsnoop. For programmers who are specifically interested in the ToolTalk communication aspects of ttsnoop, most of the ToolTalk specific code is in the following files: <code>receive.c</code>, <code>receive_pattern.c</code>, and <code>send_message.c</code>.</p> <p>Once ttsnoop is started, you can activate message reception by pressing the square "Start" button on the upper left of the main panel. ttsnoop will now display any incoming messages which match the patterns you register. To test ttsnoop, popup the "Messages..." panel and press the "Send Message" button. You should see a ToolTalk message appear on the main text subwindow of ttsnoop.</p> <p>The main panel of ttsnoop has several buttons:</p> <p>Pressing the "Messages..." button will popup a panel which allows you to compose and send ToolTalk messages. Refer to the ToolTalk Reference Manual to understand what the myriad of choices mean. Note that sending message contexts requires pressing the "Edit Contexts" button to bring up another popup, which will allow you to edit contexts to be sent with your messages.</p> <p>Pressing the "Patterns..." button will popup a panel which allows you to compose and register ToolTalk patterns. Refer to the ToolTalk Reference Manual to understand what the myriad of choices mean. By registering appropriate patterns, a programmer can use ttsnoop as a debugging tool to observe what messages are being sent by other applications. Note that using message contexts in patterns requires pressing the "Edit Contexts" button to bring up another popup, which will allow you to edit contexts to be registered with your patterns.</p> <p>Pressing the "Display..." button will popup a panel of checkboxes, each of which controls the highlighting of a specific ToolTalk message component on the ttsnoop display subwindow. Selected components will be indicated on displayed messages by a "--->" to the left of the displayed message component.</p> <p>Using the "Send Message" button allows you send one of a number of pre-selected messages, created by the "Add Message" button on the "Messages..." popup panel.</p> |

OPTIONS

-t Print trace output. Of particular interest is that ttsnoop will print example ToolTalk API code in the invoking OpenWindows cmdtool (or console if ttsnoop is invoked via a menu choice) showing what ToolTalk API calls are being used to construct a particular pattern or message.

SEE ALSO

ttsession(1)

DIAGNOSTICS

If you try and invoke ttsnoop (or any ToolTalk application) and you get a message saying the application could not start ToolTalk, or ttsession, make sure that you have one of the environment variables DISPLAY or _SUN_TT_SESSION set, and that ttsession is in your PATH, or that the SUN_TTSESSION_CMD environment variable indicates where the ttsession program resides. For more information on ttsession and the environment variables it uses, see the ttsession man page.

| | |
|---------------------------|--|
| NAME | tttar – (de)archive files and ToolTalk objects |
| SYNOPSIS | <pre>tttar c t x [EfhpSv] [<i>tarfile</i>] <i>pathname1</i> <i>pathname2</i> ... tttar c t x fL [EhpRSv] <i>tttarfile</i> [[-rename <i>oldname newname</i>] ...] <i>pathname1</i> <i>pathname2</i> ... tttar -v tttar -h[elp]</pre> |
| DESCRIPTION | <p>tttar has two fundamentally different modes. Without the L function modifier, tttar acts as a ToolTalk-aware wrapper for tar(1), archiving (or extracting) multiple files and their TT objects onto (or from) a single archive, called a <i>tarfile</i>. With the L function modifier, tttar does not invoke tar to archive actual files, but instead archives (or extracts) only TT objects onto (or from) a single archive, called a <i>tttarfile</i>. Since without the L function modifier tttar acts like an TT-aware tar(1), the discussion below will be phrased as if the L function modifier is in effect. That is, we will refer to <i>tttarfiles</i> instead of <i>tarfiles</i>, and we will discuss (de)archiving only "the TT objects of the named files" rather than (de)archiving both "the named files <i>and</i> their TT objects".</p> <p>tttar's actions are controlled by the first argument, the <i>key</i>, a string of characters containing exactly one function letter from the set ctx, and one or more of the optional function modifiers listed below. Other arguments to tttar are file or directory names that specify which files to archive or extract TT objects for. By default, the appearance of a directory name refers recursively to the files and subdirectories of that directory.</p> |
| FUNCTION LETTERS | <p>c Create a new <i>tttarfile</i> and write the TT objects of the named files onto it.</p> <p>t List the table of contents of the <i>tttarfile</i>.</p> <p>x Extract the TT objects of the named files from the <i>tttarfile</i>. If a named file matches a directory with contents written onto the tape, this directory is (recursively) extracted. The owner and modification time of the TT objects are restored (if possible). If no <i>filename</i> arguments are given, the TT objects of all files named in the archive are extracted.</p> |
| FUNCTION MODIFIERS | <p>f Use the next argument as the name of the <i>tttarfile</i>. If <i>tttarfile</i> is given as '-', tttar writes to the standard output or reads from the standard input, whichever is appropriate.</p> <p>h Follow symbolic links as if they were normal files or directories. Normally, tttar does not follow symbolic links.</p> <p>p Preserve. Restore the named files to their original modes, ignoring the present umask(2). SetUID and sticky information are also extracted if you are the super-user. This option is only useful with the x function letter, and has no meaning if the L function letter is given.</p> <p>L Do not invoke tar(1). Must be used with the f function modifier, since reading and writing an tttar archive directly to or from magnetic tape is unimplemented.</p> |

| | |
|-----------------|---|
| R | Do not recurse into directories. Valid only with the L function modifier. |
| v | Verbose. Normally tttar does its work silently; this option displays the name of each file tttar treats, preceded by the function letter. |
| OPTIONS | <p>-rename Takes the next two arguments as an <i>oldname</i> and a <i>newname</i>, respectively, and renames any entry archived as <i>oldname</i> to <i>newname</i>. If <i>oldname</i> is a directory, then its entries are recursively renamed, as well. If more than one -rename option applies to an entry (because of 1 or more parent directories being renamed), the most specific -rename option applies. The -rename option is only allowed with the L function modifier.</p> <p>-h Print usage information for tttar.</p> <p>-v Print the version of tttar.</p> |
| FILES | <p>/mountpoint/TT_DB This directory is used as a database for the ToolTalk objects of files in the filesystem mounted at <i>/mountpoint</i>.</p> |
| SEE ALSO | tar(1) , ttcp(1) , ttsession(1) , ttdbck(1M) |
| BUGS | <p>When extracting from a tar archive that is given to tttar either on magnetic tape or on the standard input, the current working directory must be writable, so that the <i>tttarfile</i> can be placed there temporarily.</p> <p>Writing and reading <i>tttarfiles</i> (i.e., archives produced with the L function modifier) directly to and from magnetic tape is unimplemented.</p> <p>The r and u function letters of tar(1), for incrementally updating an archive, are not supported.</p> <p>The X and F function modifiers and the -I option of tar(1), for including or excluding files from being archived based on SCCS status or being listed in a special file, are not supported.</p> <p>The w function modifier and the -C option of tar(1), for pausing or changing directories between the files listed on the command line, are not supported.</p> <p>A file does not have to exist for an TT object to be associated with its pathname. When tttar descends into a directory, it does not attempt to archive the objects associated with any files that do not exist in the directory.</p> |

| | |
|--------------------|--|
| NAME | tt_type_comp – the ToolTalk otype and ptype compiler |
| SYNOPSIS | <pre> tt_type_comp [-s] [-d <i>db</i>] [-mM] <i>source_file</i> tt_type_comp [-s] [-d <i>db</i>] -r <i>type</i> ... tt_type_comp [-sE] -p O P [-d <i>db</i>] tt_type_comp [-s] -p O P <i>source_file</i> tt_type_comp [-s] -x [-o <i>compiled_file</i>] <i>source_file</i> tt_type_comp [-hv] </pre> |
| DESCRIPTION | <p>tt_type_comp processes otypes and ptypes, which are descriptions of ToolTalk objects and ToolTalk processes. The default action of tt_type_comp is to compile types from source form into compiled form and then merge the compiled types into the standard ToolTalk types databases. tt_type_comp preprocesses the source types with cpp(1), and can optionally write out the compiled types instead of merging them into the standard databases. tt_type_comp can also remove types from the standard databases, or print the contents of these databases.</p> <p>tt_type_comp operates in two fundamental modes: XDR and Classing Engine. XDR mode is the default. In XDR mode, the standard databases are simply serialized ToolTalk data structures, and the format of tt_type_comp output files is the same as that of the databases. In Classing Engine mode, the standard databases are in fact the Classing Engine's own databases, and the format of tt_type_comp output files is that expected for input to ce_db_build(1) and ce_db_merge(1).</p> |
| OPTIONS | <p>-E Use Classing Engine mode, instead of the default XDR mode.</p> <p>-d Specify the database to work on which should be one of user, system, or network. For Classing Engine mode these are defined as:</p> <pre> user -- \$HOME/.cetables/cetables system -- /etc/cetables/cetables network -- \$OPENWINHOME/lib/cetables/cetables </pre> <p>For the XDR format these are defined as:</p> <pre> user -- \$HOME/.tt/types.xdr system -- /etc/tt/types.xdr network -- \$OPENWINHOME/etc/tt/types.xdr </pre> <p>These three databases form a hierarchy in which the definition of a type in the user database overrides the definition in the system database, and so on. For the merge and remove options, the default database is user. For the print options, the default is all three databases.</p> <p>-h Print help on invoking tt_type_comp and exit.</p> <p>-m Merge types into specified database, updating any existing type with the new definition given. This is the default action. This action is no longer supported for Classing Engine mode.</p> |

- M** Merge types into specified database, but only if they do not already exist in that database. This action is no longer supported for Classing Engine mode.
- p** Pretty-print on stdout all the ToolTalk types read. Types are printed out in source format suitable for recompilation with **tt_type_comp**.
- O** Enumerate on stdout the names of all otypes read.
- P** Enumerate on stdout the names of all ptypes read.
- o** *compiled_file*
With the **-x** (compile) option, emit the compiled types into the specified file, or stdout if *compiled_file* is "-".
- source_file*
Read source types from the specified file, or stdin if *source_file* is "-".
- r** *type...*
Remove the given ptypes or otypes from the specified database.
- s** Silent mode. Don't print anything.
- x** Compile source types into a compiled types file, instead of merging them into the standard types databases.
- v** Print out the version number and exit.

ENVIRONMENT**TTPATH**

In XDR mode, a colon-separated list of directories that tells ToolTalk where to find the ToolTalk types databases. The format of this variable is userDir[:systemDir[:networkDir]].

CEPATH

In Classing Engine mode, a colon-separated list of directories that tells the Classing Engine where to find the databases that contain (among other things) ToolTalk types. See **ce_db_build(1)**.

FILES

\$HOME/.tt/types.xdr User's ToolTalk types database for XDR mode

/etc/tt/types.xdr System ToolTalk types database for XDR mode

\$OPENWINHOME/etc/tt/types.xdr

Network ToolTalk types database for XDR mode

\$HOME/.cetables/cetables

/etc/cetables/cetables

\$OPENWINHOME/lib/cetables/cetables

Classing Engine databases containing ToolTalk types for CE mode. See **ce_db_build(1)**.

SEE ALSO

ttsession(1), **ce_db_build(1)**, **ce_db_merge(1)** **cpp(1)**

DIAGNOSTICS

tt_type_comp exits with the following exit codes:

- 0** Normal termination.
- 1** Usage. **tt_type_comp** was given invalid command line options.
- 2** Syntax error. A syntax error was found in the source types given to **tt_type_comp**.
- 3** System error. **tt_type_comp** was interrupted by SIGINT, or encountered some system or internal error.

| | |
|--------------------|---|
| NAME | twm – Tab Window Manager for the X Window System |
| SYNOPSIS | twm [-display <i>dpy</i>] [-s] [-f <i>initfile</i>] [-v] |
| DESCRIPTION | <p>The twm program is a window manager for the X Window System. It provides titlebars, shaped windows, several forms of icon management, user-defined macro functions, click-to-type and pointer-driven keyboard focus, and user-specified key and pointer button bindings.</p> <p>This program is usually started by the user's session manager or startup script. When used from xdm(1) or xinit(1) without a session manager, twm is frequently executed in the foreground as the last client. When run this way, exiting twm causes the session to be terminated (i.e. logged out).</p> <p>By default, application windows are surrounded by a "frame" with a titlebar at the top and a special border around the window. The titlebar contains the window's name, a rectangle that is lit when the window is receiving keyboard input, and function boxes known as "titlebuttons" at the left and right edges of the titlebar.</p> <p>Pressing pointer Button1 (usually the left-most button unless it has been changed with xmodmap(1)) on a titlebutton will invoke the function associated with the button. In the default interface, windows are iconified by clicking (pressing and then immediately releasing) the left titlebutton (which looks like a Dot). Conversely, windows are deiconified by clicking in the associated icon or entry in the icon manager (see description of the variable ShowIconManager and of the function f.showiconmgr).</p> <p>Windows are resized by pressing the right titlebutton (which resembles a group of nested squares), dragging the pointer over edge that is to be moved, and releasing the pointer when the outline of the window is the desired size. Similarly, windows are moved by pressing in the title or highlight region, dragging a window outline to the new location, and then releasing when the outline is in the desired position. Just clicking in the title or highlight region raises the window without moving it.</p> <p>When new windows are created, twm will honor any size and location information requested by the user (usually through -geometry command line argument or resources for the individual applications). Otherwise, an outline of the window's default size, its titlebar, and lines dividing the window into a 3x3 grid that track the pointer are displayed. Clicking pointer Button1 will position the window at the current position and give it the default size. Pressing pointer Button2 (usually the middle pointer button) and dragging the outline will give the window its current position but allow the sides to be resized as described above. Clicking pointer Button3 (usually the right pointer button) will give the window its current position but attempt to make it long enough to touch the bottom the screen.</p> |
| OPTIONS | <p>twm accepts the following command line options:</p> <p>-display <i>dpy</i> This option specifies the X server to use.</p> |

- s This option indicates that only the default screen (as specified by –display or by the DISPLAY environment variable) should be managed. By default, twm will attempt to manage all screens on the display.
- f *filename*
This option specifies the name of the startup file to use. By default, twm will look in the user's home directory for files named .twmrc.*num* (where *num* is a screen).
- v This option indicates that twm should print error messages whenever an unexpected X Error event is received. This can be useful when debugging applications but can be distracting in regular use.

CUSTOMIZATION

Much of twm's appearance and behavior can be controlled by providing a startup file in one of the following locations (searched in order for each screen being managed when twm begins):

\$HOME/.twmrc.*screennumber*

The *screennumber* is a small positive number (e.g. 0, 1, etc.) representing the screen number (e.g. the last number in the DISPLAY environment variable *host:displaynum.screennum*) that would be used to contact that screen of the display. This is intended for displays with multiple screens of differing visual types.

\$HOME/.twmrc

This is the usual name for an individual user's startup file.

/usr/openwin/lib/X11/system.twmrc

If neither of the preceding files are found, twm will look in this file for a default configuration. This is often tailored by the site administrator to provide convenient menus or familiar bindings for novice users.

If no startup files are found, twm will use the built-in defaults described above. The only resource used by twm is **bitmapFilePath** for a colon-separated list of directories to search when looking for bitmap files (for more information, see the *Athena Widgets* manual and **xrdb(1)**).

twm startup files are logically broken up into three types of specifications: **Variables**, **Bindings**, **Menus**. The **Variables** section must come first and is used to describe the fonts, colors, cursors, border widths, icon and window placement, highlighting, autoraising, layout of titles, warping, use of the icon manager. The **Bindings** section usually comes second and is used to specify the functions that should be to be invoked when keyboard and pointer buttons are pressed in windows, icons, titles, and frames. The **Menus** section gives any user-defined menus (containing functions to be invoked or commands to be executed).

Variable names and keywords are case-insensitive. Strings must be surrounded by double quote characters (e.g. "blue") and are case-sensitive. A pound sign (#) outside of a string causes the remainder of the line in which the character appears to be treated as a comment.

VARIABLES

Many of the aspects of **twm**'s user interface are controlled by variables that may be set in the user's startup file. Some of the options are enabled or disabled simply by the presence of a particular keyword. Other options require keywords, numbers, strings, or lists of all of these.

Lists are surrounded by braces and are usually separated by whitespace or a newline. For example:

```
AutoRaise { "emacs" "XTerm" "Xmh" }
```

or

```
AutoRaise
{
    "emacs"
    "XTerm"
    "Xmh"
}
```

When a variable containing a list of strings representing windows is searched (e.g. to determine whether or not to enable autoraise as shown above), a string must be an exact, case-sensitive match to the window's name (given by the **WM_NAME** window property), resource name or class name (both given by the **WM_CLASS** window property). The preceding example would enable autoraise on windows named "emacs" as well as any **xterm**(1) (since they are of class "XTerm") or **xmh** windows (which are of class "Xmh").

String arguments that are interpreted as filenames (see the **Pixmap**, **Cursors**, and **Icon-Directory** below) will prepend the user's directory (specified by the **HOME** environment variable) if the first character is a tilde (~). If, instead, the first character is a colon (:), the name is assumed to refer to one of the internal bitmaps that are used to create the default titlebars symbols: **:xlogo** or **:iconify** (both refer to the X used for the iconify button), **:resize** (the nested squares used by the resize button), and **:question** (the question mark used for non-existent bitmap files).

The following variables may be specified at the top of a **twm** startup file. Lists of Window name prefix strings are indicated by *win-list*. Optional arguments are shown in square brackets:

AutoRaise { *win-list* }

This variable specifies a list of windows that should automatically be raised whenever the pointer enters the window. This action can be interactively enabled or disabled on individual windows using the function **f.autoraise**.

AutoRelativeResize

This variable indicates that dragging out a window size (either when initially sizing the window with pointer Button2 or when resizing it) should not wait until the pointer has crossed the window edges. Instead, moving the pointer automatically causes the nearest edge or edges to move by the same amount. This allows the resizing of windows that extend off the edge of the screen. If the pointer is in the center of the window, or if the resize is begun by pressing a

titlebutton, **twm** will still wait for the pointer to cross a window edge (to prevent accidents). This option is particularly useful for people who like the press-drag-release method of sweeping out window sizes.

BorderColor *string* [{ *wincolorlist* }]

This variable specifies the default color of the border to be placed around all non-iconified windows, and may only be given within a **Color** or **Monochrome** list. The optional *wincolorlist* specifies a list of window and color name pairs for specifying particular border colors for different types of windows. For example:

```
BorderColor "gray50"
{
    "XTerm"      "red"
    "xmh"       "green"
}
```

The default is "black".

BorderTileBackground *string* [{ *wincolorlist* }]

This variable specifies the default background color in the gray pattern used in unhighlighted borders (only if **NoHighlight** hasn't been set), and may only be given within a **Color** or **Monochrome** list. The optional *wincolorlist* allows per-window colors to be specified. The default is "white".

BorderTileForeground *string* [{ *wincolorlist* }]

This variable specifies the default foreground color in the gray pattern used in unhighlighted borders (only if **NoHighlight** hasn't been set), and may only be given within a **Color** or **Monochrome** list. The optional *wincolorlist* allows per-window colors to be specified. The default is "black".

BorderWidth *pixels*

This variable specifies the width in pixels of the border surrounding all client window frames if **ClientBorderWidth** has not been specified. This value is also used to set the border size of windows created by **twm** (such as the icon manager). The default is 2.

ButtonIndent *pixels*

This variable specifies the amount by which titlebuttons should be indented on all sides. Positive values cause the buttons to be smaller than the window text and highlight area so that they stand out. Setting this and the **TitleButtonBorderWidth** variables to 0 makes titlebuttons be as tall and wide as possible. The default is 1.

ClientBorderWidth

This variable indicates that border width of a window's frame should be set to the initial border width of the window, rather than to the value of **BorderWidth**.

Color { *colors-list* }

This variable specifies a list of color assignments to be made if the default display is capable of displaying more than simple black and white. The *colors-list* is made up of the following color variables and their values:

DefaultBackground, DefaultForeground, MenuBackground, MenuForeground, MenuTitleBackground, MenuTitleForeground, and MenuShadowColor. The following color variables may also be given a list of window and color name pairs to allow per-window colors to be specified (see **BorderColor** for details): **BorderColor, IconManagerHighlight, BorderTitleBackground, BorderTitleForeground, TitleBackground, TitleForeground, IconBackground, IconForeground, IconBorderColor, IconManagerBackground,** and **IconManagerForeground.** For example:

```
Color
{
    MenuBackground          "gray50"
    MenuForeground          "blue"
    BorderColor             "red" { "XTerm" "yellow" }
    TitleForeground         "yellow"
    TitleBackground         "blue"
}
```

All of these color variables may also be specified for the **Monochrome** variable, allowing the same initialization file to be used on both color and monochrome displays.

ConstrainedMoveTime *milliseconds*

This variable specifies the length of time between button clicks needed to begin a constrained move operation. Double clicking within this amount of time when invoking **f.move** will cause the window only be moved in a horizontal or vertical direction. Setting this value to 0 will disable constrained moves. The default is 400 milliseconds.

Cursors { *cursor-list* }

This variable specifies the glyphs that **twm** should use for various pointer cursors. Each cursor may be defined either from the **cursor** font or from two bitmap files. Shapes from the **cursor** font may be specified directly as:

```
cursorname    "string"
```

where *cursorname* is one of the cursor names listed below, and *string* is the name of a glyph as found in the file `/usr/include/X11/cursorfont.h` (without the "XC_" prefix). If the cursor is to be defined from bitmap files, the following syntax is used instead:

```
cursorname    "image" "mask"
```

The *image* and *mask* strings specify the names of files containing the glyph image and mask in **bitmap(1)** form. The bitmap files are located in the same manner as

icon bitmap files. The following example shows the default cursor definitions:

```

Cursors
{
    Frame           "top_left_arrow"
    Title           "top_left_arrow"
    Icon            "top_left_arrow"
    IconMgr         "top_left_arrow"
    Move            "fleur"
    Resize          "fleur"
    Menu            "sb_left_arrow"
    Button          "hand2"
    Wait            "watch"
    Select          "dot"
    Destroy "pirate"
}

```

DecorateTransients

This variable indicates that transient windows (those containing a WM_TRANSIENT_FOR property) should have titlebars. By default, transients are not reparented.

DefaultBackground *string*

This variable specifies the background color to be used for sizing and information windows. The default is "white".

DefaultForeground *string*

This variable specifies the foreground color to be used for sizing and information windows. The default is "black".

DontIconifyByUnmapping { *win-list* }

This variable specifies a list of windows that should not be iconified by simply unmapping the window (as would be the case if **IconifyByUnmapping** had been set). This is frequently used to force some windows to be treated as icons while other windows are handled by the icon manager.

DontMoveOff

This variable indicates that windows should not be allowed to be moved off the screen. It can be overridden by the **f.forcemove** function.

DontSqueezeTitle [{ *win-list* }]

This variable indicates that titlebars should not be squeezed to their minimum size as described under **SqueezeTitle** below. If the optional window list is supplied, only those windows will be prevented from being squeezed.

ForceIcons

This variable indicates that icon pixmaps specified in the **Icons** variable should override any client-supplied pixmaps.

FramePadding *pixels*

This variable specifies the distance between the titlebar decorations (the button

and text) and the window frame. The default is 2 pixels.

IconBackground *string* [{ *win-list* }]

This variable specifies the background color of icons, and may only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. See the **BorderColor** variable for a complete description of the *win-list*. The default is "white".

IconBorderColor *string* [{ *win-list* }]

This variable specifies the color of the border used for icon windows, and may only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. See the **BorderColor** variable for a complete description of the *win-list*. The default is "black".

IconBorderWidth *pixels*

This variable specifies the width in pixels of the border surrounding icon windows. The default is 2.

IconDirectory *string*

This variable specifies the directory that should be searched if a bitmap file cannot be found in any of the directories in the **bitmapFilePath** resource.

IconFont *string*

This variable specifies the font to be used to display icon names within icons. The default is "variable".

IconForeground *string* [{ *win-list* }]

This variable specifies the foreground color to be used when displaying icons, and may only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. See the **BorderColor** variable for a complete description of the *win-list*. The default is "black".

IconifyByUnmapping [{ *win-list* }]

This variable indicates that windows should be iconified by being unmapped without trying to map any icons. This assumes that the user will remap the window through the icon manager, the **f.warpto** function, or the *TwmWindows* menu. If the optional *win-list* is provided, only those windows will be iconified by simply unmapping. Windows that have both this and the **IconManagerDontShow** options set may not be accessible if no binding to the *TwmWindows* menu is set in the user's startup file.

IconManagerBackground *string* [{ *win-list* }]

This variable specifies the background color to use for icon manager entries, and may only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. See the **BorderColor** variable for a complete description of the *win-list*. The default is "white".

IconManagerDontShow [{ *win-list* }]

This variable indicates that the icon manager should not display any windows.

If the optional *win-list* is given, only those windows will not be displayed. This variable is used to prevent windows that are rarely iconified (such as **xclock(1)** or **xload(1)**) from taking up space in the icon manager.

IconManagerFont *string*

This variable specifies the font to be used when displaying icon manager entries. The default is "variable".

IconManagerForeground *string* [{ *win-list* }]

This variable specifies the foreground color to be used when displaying icon manager entries, and may only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. See the **BorderColor** variable for a complete description of the *win-list*. The default is "black".

IconManagerGeometry *string* [*columns*]

This variable specifies the geometry of the icon manager window. The *string* argument is standard geometry specification that indicates the initial full size of the icon manager. The icon manager window is then broken into *columns* pieces and scaled according to the number of entries in the icon manager. Extra entries are wrapped to form additional rows. The default number of columns is 1.

IconManagerHighlight *string* [{ *win-list* }]

This variable specifies the border color to be used when highlighting the icon manager entry that currently has the focus, and can only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. See the **BorderColor** variable for a complete description of the *win-list*. The default is "black".

IconManagers { *iconmgr-list* }

This variable specifies a list of icon managers to create. Each item in the *iconmgr-list* has the following format:

```
"winname" ["iconname"] "geometry" columns
```

where *winname* is the name of the windows that should be put into this icon manager, *iconname* is the name of that icon manager window's icon, *geometry* is a standard geometry specification, and *columns* is the number of columns in this icon manager as described in **IconManagerGeometry**. For example:

```
IconManagers
{
    "XTerm"      "=300x5+800+5" 5
    "myhost"    "=400x5+100+5" 2
}
```

Clients whose name or class is "XTerm" will have an entry created in the "XTerm" icon manager. Clients whose name was "myhost" would be put into the "myhost" icon manager.

IconManagerShow { *win-list* }

This variable specifies a list of windows that should appear in the icon manager. When used in conjunction with the **IconManagerDontShow** variable, only the windows in this list will be shown in the icon manager.

IconRegion *geomstring vgrav hgrav gridwidth gridheight*

This variable specifies an area on the root window in which icons are placed if no specific icon location is provided by the client. The *geomstring* is a quoted string containing a standard geometry specification. If more than one **IconRegion** lines are given, icons will be put into the succeeding icon regions when the first is full. The **vgrav** argument should be either **North** or **South** and control and is used to control whether icons are first filled in from the top or bottom of the icon region. Similarly, the **hgrav** argument should be either **East** or **West** and is used to control whether icons should be filled in from left from the right. Icons are laid out within the region in a grid with cells *gridwidth* pixels wide and *gridheight* pixels high.

Icons { *win-list* }

This variable specifies a list of window names and the bitmap filenames that should be used as their icons. For example:

```
Icons
{
    "XTerm"      "xterm.icon"
    "xfd"        "xfd_icon"
}
```

Windows that match "XTerm" and would not be iconified by unmapping, and would try to use the icon bitmap in the file "xterm.icon". If **ForceIcons** is specified, this bitmap will be used even if the client has requested its own icon pixmap.

InterpolateMenuColors

This variable indicates that menu entry colors should be interpolated between entry specified colors. In the example below:

```
Menu "mymenu"
{
    "Title"      ("black":"red")      f.title
    "entry1"     f.nop
    "entry2"     f.nop
    "entry3"     ("white":"green")    f.nop
    "entry4"     f.nop
    "entry5"     ("red":"white")      f.nop
}
```

the foreground colors for "entry1" and "entry2" will be interpolated between black and white, and the background colors between red and green. Similarly, the foreground for "entry4" will be half-way between white and red, and the background will be half-way between green and white.

MakeTitle { *win-list* }

This variable specifies a list of windows on which a titlebar should be placed and is used to request titles on specific windows when **NoTitle** has been set.

MaxWindowSize *string*

This variable specifies a geometry in which the width and height give the maximum size for a given window. This is typically used to restrict windows to the size of the screen. The default is "30000x30000".

MenuBackground *string*

This variable specifies the background color used for menus, and can only be specified inside of a **Color** or **Monochrome** list. The default is "white".

MenuFont *string*

This variable specifies the font to use when displaying menus. The default is "variable".

MenuForeground *string*

This variable specifies the foreground color used for menus, and can only be specified inside of a **Color** or **Monochrome** list. The default is "black".

MenuShadowColor *string*

This variable specifies the color of the shadow behind pull-down menus and can only be specified inside of a **Color** or **Monochrome** list. The default is "black".

MenuTitleBackground *string*

This variable specifies the background color for **f.title** entries in menus, and can only be specified inside of a **Color** or **Monochrome** list. The default is "white".

MenuTitleForeground *string*

This variable specifies the foreground color for **f.title** entries in menus and can only be specified inside of a **Color** or **Monochrome** list. The default is "black".

Monochrome { *colors* }

This variable specifies a list of color assignments that should be made if the screen has a depth of 1. See the description of **Colors**.

MoveDelta *pixels*

This variable specifies the number of pixels the pointer must move before the **f.move** function starts working. Also see the **f.deltastop** function. The default is zero pixels.

NoBackingStore

This variable indicates that **twm**'s menus should not request backing store to minimize repainting of menus. This is typically used with servers that can repaint faster than they can handle backing store.

NoCaseSensitive

This variable indicates that case should be ignored when sorting icon names in an icon manager. This option is typically used with applications that capitalize the first letter of their icon name.

NoDefaults

This variable indicates that **twm** should not supply the default titlebuttons and

bindings. This option should only be used if the startup file contains a completely new set of bindings and definitions.

NoGrabServer

This variable indicates that **twm** should not grab the server when popping up menus and moving opaque windows.

NoHighlight [{ *win-list* }]

This variable indicates that borders should not be highlighted to track the location of the pointer. If the optional *win-list* is given, highlighting will only be disabled for those windows. When the border is highlighted, it will be drawn in the current **BorderColor**. When the border is not highlighted, it will be stippled with a gray pattern using the current **BorderTileForeground** and **BorderTileBackground** colors.

NoIconManagers

This variable indicates that no icon manager should be created.

NoMenuShadows

This variable indicates that menus should not have drop shadows drawn behind them. This is typically used with slower servers since it speeds up menu drawing at the expense of making the menu slightly harder to read.

NoRaiseOnDeiconify

This variable indicates that windows that are deiconified should not be raised.

NoRaiseOnMove

This variable indicates that windows should not be raised when moved. This is typically used to allow windows to slide underneath each other.

NoRaiseOnResize

This variable indicates that windows should not be raised when resized. This is typically used to allow windows to be resized underneath each other.

NoRaiseOnWarp

This variable indicates that windows should not be raised when the pointer is warped into them with the **f.warpto** function. If this option is set, warping to an occluded window may result in the pointer ending up in the occluding window instead the desired window (which causes unexpected behavior with **f.warpring**).

NoSaveUnders

This variable indicates that menus should not request save-unders to minimize window repainting following menu selection. It is typically used with displays that can repaint faster than they can handle save-unders.

NoStackMode [{ *win-list* }]

This variable indicates that client window requests to change stacking order should be ignored. If the optional *win-list* is given, only requests on those windows will be ignored. This is typically used to prevent applications from relentlessly popping themselves to the front of the window stack.

NoTitle [{ *win-list* }]

This variable indicates that windows should not have titlebars. If the optional *win-list* is given, only those windows will not have titlebars. **MakeTitle** may be used with this option to force titlebars to be put on specific windows.

NoTitleFocus

This variable indicates that **twm** should not set keyboard input focus to each window as it is entered. Normally, **twm** sets the focus so that focus and key events from the titlebar and icon managers are delivered to the application. If the pointer is moved quickly and **twm** is slow to respond, input can be directed to the old window instead of the new. This option is typically used to prevent this "input lag" and to work around bugs in older applications that have problems with focus events.

NoTitleHighlight [{ *win-list* }]

This variable indicates that the highlight area of the titlebar, which is used to indicate the window that currently has the input focus, should not be displayed. If the optional *win-list* is given, only those windows will not have highlight areas. This and the **SqueezeTitle** options can be set to substantially reduce the amount of screen space required by titlebars.

OpaqueMove

This variable indicates that the **f.move** function should actually move the window instead of just an outline so that the user can immediately see what the window will look like in the new position. This option is typically used on fast displays (particularly if **NoGrabServer** is set).

Pixmaps { *pixmap*s }

This variable specifies a list of pixmaps that define the appearance of various images. Each entry is a keyword indicating the pixmap to set, followed by a string giving the name of the bitmap file. The following pixmaps may be specified:

```

Pixmaps
{
    TitleHighlight "gray1"
}

```

The default for *TitleHighlight* is to use an even stipple pattern.

RandomPlacement

This variable indicates that windows with no specified geometry should be placed in a pseudo-random location instead of having the user drag out an outline.

ResizeFont *string*

This variable specifies the font to be used for in the dimensions window when resizing windows. The default is "fixed".

RestartPreviousState

This variable indicates that **twm** should attempt to use the WM_STATE property on client windows to tell which windows should be iconified and which

should be left visible. This is typically used to try to regenerate the state that the screen was in before the previous window manager was shutdown.

SaveColor { *colors-list* }

This variable indicates a list of color assignments to be stored as pixel values in the root window property `_MIT_PRIORITY_COLORS`. Clients may elect to preserve these values when installing their own colormap. Note that use of this mechanism is a way an for application to avoid the "technicolor" problem, whereby useful screen objects such as window borders and titlebars disappear when a programs custom colors are installed by the window manager. For example:

```

SaveColor
{
    BorderColor
    TitleBackground
    TitleForeground
    "red"
    "green"
    "blue"
}

```

This would place on the root window 3 pixel values for borders and titlebars, as well as the three color strings, all taken from the default colormap.

ShowIconManager

This variable indicates that the icon manager window should be displayed when **twm** is started. It can always be brought up using the **f.showiconmgr** function.

SortIconManager

This variable indicates that entries in the icon manager should be sorted alphabetically rather than by simply appending new windows to the end.

SqueezeTitle [{ *squeeze-list* }]

This variable indicates that **twm** should attempt to use the SHAPE extension to make titlebars occupy only as much screen space as they need, rather than extending all the way across the top of the window. The optional *squeeze-list* may be used to control the location of the squeezed titlebar along the top of the window. It contains entries of the form:

```
"name"      justification      num      denom
```

where *name* is a window name, *justification* is either **left**, **center**, or **right**, and *num* and *denom* are numbers specifying a ratio giving the relative position about which the titlebar is justified. The ratio is measured from left to right if the numerator is positive, and right to left if negative. A denominator of 0 indicates that the numerator should be measured in pixels. For convenience, the ratio 0/0

is the same as 1/2 for **center** and -1/1 for **right**. For example:

```
SqueezeTitle
{
    "XTerm"      left      0      0
    "xterm1"    left      1      3
    "xterm2"    left      2      3
    "oclock"center  0      0
    "emacs"     right     0      0
}
```

The **DontSqueezeTitle** list can be used to turn off squeezing on certain titles.

StartIconified *string* *win-list*

This variable indicates that client windows should initially be left as icons until explicitly deiconified by the user. If the optional *win-list* is given, only those windows will be started iconic. This is useful for programs that do not support an **-iconic** command line option or resource.

TitleBackground *string* *win-list*

This variable specifies the background color used in titlebars, and may only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. The default is "white".

TitleButtonBorderWidth *pixels*

This variable specifies the width in pixels of the border surrounding titlebuttons. This is typically set to 0 to allow titlebuttons to take up as much space as possible and to not have a border. The default is 1.

TitleFont *string*

This variable specifies the font to be used for displaying window names in titlebars. The default is "variable".

TitleForeground *string* *win-list*

This variable specifies the foreground color used in titlebars, and may only be specified inside of a **Color** or **Monochrome** list. The optional *win-list* is a list of window names and colors so that per-window colors may be specified. The default is "black".

TitlePadding *pixels*

This variable specifies the distance between the various buttons, text, and highlight areas in the titlebar. The default is 8 pixels.

UnknownIcon *string*

This variable specifies the filename of a bitmap file to be used as the default icon. This bitmap will be used as the icon of all clients which do not provide an icon bitmap and are not listed in the **Icons** list.

UsePPosition *string*

This variable specifies whether or not **twm** should honor program-requested locations (given by the **PPosition** flag in the WM_NORMAL_HINTS property)

in the absence of a user-specified position. The argument *string* may have one of three values: "off" (the default) indicating that **twm** should ignore the program-supplied position, "on" indicating that the position should be used, and "non-zero" indicating that the position should be used if it is other than (0,0). The latter option is for working around a bug in older toolkits.

WarpCursor [{ *win-list* }]

This variable indicates that the pointer should be warped into windows when they are deiconified. If the optional *win-list* is given, the pointer will only be warped when those windows are deiconified.

WindowRing { *win-list* }

This variable specifies a list of windows along which the **f.warpring** function cycles.

WarpUnmapped

This variable indicates that the **f.warpto** function should deiconify any iconified windows it encounters. This is typically used to make a key binding that will pop a particular window (such as **xmh**), no matter where it is. The default is for **f.warpto** to ignore iconified windows.

XorValue *number*

This variable specifies the value to use when drawing window outlines for moving and resizing. This should be set to a value that will result in a variety of distinguishable colors when exclusive-or'ed with the contents of the user's typical screen. Setting this variable to 1 often gives nice results if adjacent colors in the default colormap are distinct. By default, **twm** will attempt to cause temporary lines to appear at the opposite end of the colormap from the graphics.

Zoom [*count*]

This variable indicates that outlines suggesting movement of a window to and from its iconified state should be displayed whenever a window is iconified or deiconified. The optional *count* argument specifies the number of outlines to be drawn. The default count is 8.

The following variables must be set after the fonts have been assigned, so it is usually best to put them at the end of the variables or beginning of the bindings sections:

DefaultFunction *function*

This variable specifies the function to be executed when a key or button event is received for which no binding is provided. This is typically bound to **f.nop**, **f.beep**, or a menu containing window operations.

WindowFunction *function*

This variable specifies the function to execute when a window is selected from the **TwmWindows** menu. If this variable is not set, the window will be deiconified and raised.

BINDINGS

After the desired variables have been set, functions may be attached to titlebuttons and key and pointer buttons. Titlebuttons may be added from the left or right side and appear in the titlebar from left-to-right according to the order in which they are specified. Key and

pointer button bindings may be given in any order.

Titlebutton specifications must include the name of the pixmap to use in the button box and the function to be invoked when a pointer button is pressed within them:

LeftTitleButton "*bitmapname*" = *function*

or

RightTitleButton "*bitmapname*" = *function*

The *bitmapname* may refer to one of the built-in bitmaps (which are scaled to match **TitleFont**) by using the appropriate colon-prefixed name described above.

Key and pointer button specifications must give the modifiers that must be pressed, over which parts of the screen the pointer must be, and what function is to be invoked. Keys are given as strings containing the appropriate keysym name; buttons are given as the keywords **Button1-Button5**:

"FP1" = *modlist* : *context* : *function*
Button1 = *modlist* : *context* : *function*

The *modlist* is any combination of the modifier names **shift**, **control**, **lock**, **meta**, **mod1**, **mod2**, **mod3**, **mod4**, or **mod5** (which may be abbreviated as **s**, **c**, **l**, **m**, **m1**, **m2**, **m3**, **m4**, **m5**, respectively) separated by a vertical bar (|). Similarly, the *context* is any combination of **window**, **title**, **icon**, **root**, **frame**, **iconmgr**, their first letters (**iconmgr** abbreviation is **m**), or **all**, separated by a vertical bar. The *function* is any of the **f.** keywords described below. For example, the default startup file contains the following bindings:

```
Button1 =      : root      : f.menu "TwmWindows"
Button1 = m    : window | icon: f.function "move-or-lower"
Button2 = m    : window | icon: f.iconify
Button3 = m    : window | icon: f.function "move-or-raise"
Button1 =      : title     : f.function "move-or-raise"
Button2 =      : title     : f.raiselower
Button1 =      : icon      : f.function "move-or-iconify"
Button2 =      : icon      : f.iconify
Button1 =      : iconmgr   : f.iconify
Button2 =      : iconmgr   : f.iconify
```

A user who wanted to be able to manipulate windows from the keyboard could use the following bindings:

```
"F1" =      : all      : f.iconify
"F2" =      : all      : f.raiselower
"F3" =      : all      : f.warpring "next"
"F4" =      : all      : f.warpto "xmh"
"F5" =      : all      : f.warpto "emacs"
"F6" =      : all      : f.colormap "next"
"F7" =      : all      : f.colormap "default"
"F20" =     : all      : f.warptoscreen "next"
```

```

"Left"      = m      : all      : f.backiconmgr
"Right"     = m | s : all      : f.forwiconmgr
"Up"        = m      : all      : f.upiconmgr
"Down"      = m | s : all      : f.downiconmgr

```

twm provides many more window manipulation primitives than can be conveniently stored in a titlebar, menu, or set of key bindings. Although a small set of defaults are supplied (unless the **NoDefaults** is specified), most users will want to have their most common operations bound to key and button strokes. To do this, **twm** associates names with each of the primitives and provides *user-defined functions* for building higher level primitives and *menus* for interactively selecting among groups of functions.

User-defined functions contain the name by which they are referenced in calls to **f.function** and a list of other functions to execute. For example:

```

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 }
Function "restore-colormap" { f.colormap "default" f.lower }

```

The function name must be used in **f.function** exactly as it appears in the function specification.

In the descriptions below, if the function is said to operate on the selected window, but is invoked from a root menu, the cursor will be changed to the **Select** cursor and the next window to receive a button press will be chosen:

! *string* This is an abbreviation for **f.exec** *string*.

f.autoraise

This function toggles whether or not the selected window is raised whenever entered by the pointer. See the description of the variable **AutoRaise**.

f.backiconmgr

This function warps the pointer to the previous column in the current icon manager, wrapping back to the previous row if necessary.

f.beep This function sounds the keyboard bell.

f.bottomzoom

This function is similar to the **f.fullzoom** function, but resizes the window to fill only the bottom half of the screen.

f.circledown

This function lowers the top-most window that occludes another window.

f.circleup

This function raises the bottom-most window that is occluded by another window.

f.colormap *string*

This function rotates the colormaps (obtained from the **WM_COLORMAP_WINDOWS** property on the window) that **twm** will display when the pointer is in this window. The argument *string* may have one of the

following values: "**next**", "**prev**", and "**default**". It should be noted here that in general, the installed colormap is determined by keyboard focus. A pointer driven keyboard focus will install a private colormap upon entry of the window owning the colormap. Using the click to type model, private colormaps will not be installed until the user presses a mouse button on the target window.

f.deiconify

This function deiconifies the selected window. If the window is not an icon, this function does nothing.

f.delete This function sends the WM_DELETE_WINDOW message to the selected window if the client application has requested it through the WM_PROTOCOLS window property. The application is supposed to respond to the message by removing the indicated window. If the window has not requested WM_DELETE_WINDOW messages, the keyboard bell will be rung indicating that the user should choose an alternative method. Note this is very different from **f.destroy**. The intent here is to delete a single window, not necessarily the entire application.

f.deltastop

This function allows a user-defined function to be aborted if the pointer has been moved more than *MoveDelta* pixels. See the example definition given for **Function "move-or-raise"** at the beginning of the section.

f.destroy

This function instructs the X server to close the display connection of the client that created the selected window. This should only be used as a last resort for shutting down runaway clients. See also **f.delete**.

f.downiconmgr

This function warps the pointer to the next row in the current icon manger, wrapping to the beginning of the next column if necessary.

f.exec *string*

This function passes the argument *string* to /bin/sh for execution. In multiscreen mode, if *string* starts a new X client without giving a display argument, the client will appear on the screen from which this function was invoked.

f.focus This function toggles the keyboard focus of the server to the selected window, changing the focus rule from pointer-driven if necessary. If the selected window already was focused, this function executes an **f.unfocus**.

f.forcemove

This function is like **f.move** except that it ignores the **DontMoveOff** variable.

f.forwiconmgr

This function warps the pointer to the next column in the current icon manager, wrapping to the beginning of the next row if necessary.

f.fullzoom

This function resizes the selected window to the full size of the display or else restores the original size if the window was already zoomed.

f.function *string*

This function executes the user-defined function whose name is specified by the argument *string*.

f.hbzoom

This function is a synonym for **f.bottomzoom**.

f.hideiconmgr

This function unmaps the current icon manager.

f.horizoom

This variable is similar to the **f.zoom** function except that the selected window is resized to the full width of the display.

f.htzoom

This function is a synonym for **f.topzoom**.

f.hzoom

This function is a synonym for **f.horizoom**.

f.iconify

This function iconifies or deiconifies the selected window or icon, respectively.

f.identify

This function displays a summary of the name and geometry of the selected window. Clicking the pointer or pressing a key in the window will dismiss it.

f.lefticonmgr

This function similar to **f.backiconmgr** except that wrapping does not change rows.

f.leftzoom

This variable is similar to the **f.bottomzoom** function but causes the selected window is only resized to the left half of the display.

f.lower This function lowers the selected window.

f.menu *string*

This function invokes the menu specified by the argument *string*. Cascaded menus may be built by nesting calls to **f.menu**.

f.move This function drags an outline of the selected window (or the window itself if the **OpaqueMove** variable is set) until the invoking pointer button is released. Double clicking within the number of milliseconds given by **ConstrainedMoveTime** warps the pointer to the center of the window and constrains the move to be either horizontal or vertical depending on which grid line is crossed. To abort a move, press another button before releasing the first button.

f.nexticonmgr

This function warps the pointer to the next icon manager containing any windows on the current or any succeeding screen.

f.nop This function does nothing and is typically used with the **DefaultFunction** or **WindowFunction** variables or to introduce blank lines in menus.

f.previceiconmgr

This function warps the pointer to the previous icon manager containing any windows on the current or preceding screens.

f.quit This function causes **twm** to restore the window's borders and exit. If **twm** is the first client invoked from **xdm**, this will result in a server reset.

f.raise This function raises the selected window.

f.raiselower

This function raises the selected window to the top of the stacking order if it is occluded by any windows, otherwise the window will be lowered.

f.refresh

This function causes all windows to be refreshed.

f.resize This function displays an outline of the selected window. Crossing a border (or setting **AutoRelativeResize**) will cause the outline to begin to rubber band until the invoking button is released. To abort a resize, press another button before releasing the first button.

f.restart This function kills and restarts **twm**.

f.righticonmgr

This function is similar to **f.nexticonmgr** except that wrapping does not change rows.

f.rightzoom

This variable is similar to the **f.bottomzoom** function except that the selected window is only resized to the right half of the display.

f.saveyourself

This function sends a WM_SAVEYOURSELF message to the selected window if it has requested the message in its WM_PROTOCOLS window property. Clients that accept this message are supposed to checkpoint all state associated with the window and update the WM_COMMAND property as specified in the ICCCM. If the selected window has not selected for this message, the keyboard bell will be rung.

f.showiconmgr

This function maps the current icon manager.

f.sorticonmgr

This function sorts the entries in the current icon manager alphabetically. See the variable **SortIconManager**.

f.title This function provides a centered, unselectable item in a menu definition. It should not be used in any other context.

f.topzoom

This variable is similar to the **f.bottomzoom** function except that the selected window is only resized to the top half of the display.

f.unfocus

This function resets the focus back to pointer-driven. This should be used when a focused window is no longer desired.

f.upiconmgr

This function warps the pointer to the previous row in the current icon manager, wrapping to the last row in the same column if necessary.

f.vlzoom

This function is a synonym for **f.leftzoom**.

f.vrzoom

This function is a synonym for **f.rightzoom**.

f.warping *string*

This function warps the pointer to the next or previous window (as indicated by the argument **string**, which may be "next" or "prev") specified in the **WindowRing** variable.

f.warpto *string*

This function warps the pointer to the window which has a name or class that matches *string*. If the window is iconified, it will be deiconified if the variable **WarpUnmapped** is set or else ignored.

f.warptoiconmgr *string*

This function warps the pointer to the icon manager entry associated with the window containing the pointer in the icon manager specified by the argument *string*. If *string* is empty (i.e. ""), the current icon manager is chosen.

f.warptoscreen *string*

This function warps the pointer to the screen specified by the argument *string*. **String** may be a number (e.g. "0" or "1"), the word "next" (indicating the current screen plus 1, skipping over any unmanaged screens), the word "back" (indicating the current screen minus 1, skipping over any unmanaged screens), or the word "prev" (indicating the last screen visited).

f.winrefresh

This function is similar to the **f.refresh** function except that only the selected window is refreshed.

f.zoom This function is similar to the **f.fullzoom** function, except that the only the height of the selected window is changed.

MENUS

Functions may be grouped and interactively selected using pop-up (when bound to a pointer button) or pull-down (when associated with a titlebutton) menus. Each menu specification contains the name of the menu as it will be referred to by **f.menu**, optional default foreground and background colors, the list of item names and the functions they should invoke, and optional foreground and background colors for individual items:

```
Menu "menuname" [ ("deffore":"defback") ]
{
    string1 [ ("fore1":"backn")]    function1
    string2 [ ("fore2":"backn")]    function2
    .
    .
    .
```

```

        stringN [ ("foreN":"backN")]    functionN
    }

```

The *menuname* is case-sensitive. The optional *deffore* and *defback* arguments specify the foreground and background colors used on a color display to highlight menu entries. The *string* portion of each menu entry will be the text which will appear in the menu. The optional *fore* and *back* arguments specify the foreground and background colors of the menu entry when the pointer is not in the entry. These colors will only be used on a color display. The default is to use the colors specified by the **MenuForeground** and **MenuBackground** variables. The *function* portion of the menu entry is one of the functions, including any user-defined functions, or additional menus.

There is a special menu named **TwmWindows** which contains the names of all of the client and **twm**-supplied windows. Selecting an entry will cause the **WindowFunction** to be executed on that window. If **WindowFunction** hasn't been set, the window will be deiconified and raised.

ICONS

twm supports several different ways of manipulating iconified windows. The common pixmap-and-text style may be laid out by hand or automatically arranged as described by the **IconRegion** variable. In addition, a terse grid of icon names, called an icon manager, provides a more efficient use of screen space as well as the ability to navigate among windows from the keyboard.

An icon manager is a window that contains names of selected or all windows currently on the display. In addition to the window name, a small button using the default iconify symbol will be displayed to the left of the name when the window is iconified. By default, clicking on an entry in the icon manager performs **f.iconify**. To change the actions taken in the icon manager, use the **iconmgr** context when specifying button and keyboard bindings.

Moving the pointer into the icon manager also directs keyboard focus to the indicated window (setting the focus explicitly or else sending synthetic events **NoTitleFocus** is set). Using the **f.upiconmgr**, **f.downiconmgr**, **f.lefticonmgr**, and **f.righticonmgr** functions, the input focus can be changed between windows directly from the keyboard.

BUGS

The resource manager should have been used instead of all of the window lists.

The **IconRegion** variable should take a list.

Double clicking very fast to get the constrained move function will sometimes cause the window to move, even though the pointer is not moved.

If **IconifyByUnmapping** is on and windows are listed in **IconManagerDontShow** but not in **DontIconifyByUnmapping**, they may be lost if they are iconified and no bindings to **f.menu "TwmWindows"** or **f.warpto** are setup.

FILES

```

$HOME/.twmrc.<screen number>
$HOME/.twmrc
/usr/openwin/lib/X11/system.twmrc

```

**ENVIRONMENT
VARIABLES****DISPLAY**

This variable is used to determine which X server to use. It is also set during **f.exec** so that programs come up on the proper screen.

HOME This variable is used as the prefix for files that begin with a tilde and for locating the **twm** startup file.

SEE ALSO

X11(7), **xdm(1)**, **xrdb(1)**

COPYRIGHT

Portions copyright 1988 Evans & Sutherland Computer Corporation; portions copyright 1989 Hewlett-Packard Company and the Massachusetts Institute of Technology. See **X11(7)** for a full statement of rights and permissions.

AUTHORS

Tom LaStrange, Solbourne Computer; Jim Fulton, MIT X Consortium; Steve Pitschke, Stardent Computer; Keith Packard, MIT X Consortium; Dave Sternlicht, MIT X Consortium; Dave Payne, Apple Computer.

| | |
|--------------------|--|
| NAME | viewprint – print AnswerBook documents using lp |
| SYNOPSIS | viewprint -P <i>printer</i> / -f <i>file</i> [-# <i>ncopies</i>] [-p <i>pages</i>] [-R] [-c <i>card-catalog-file</i>] <i>document-name</i> |
| DESCRIPTION | <p>viewprint is a utility used by docviewer(1) to print selected pages of AnswerBook documents. viewprint uses the AnswerBook "card catalog" mechanism to locate the PostScript files corresponding to the specified document <i>document-name</i>, and prints these files using lp(1).</p> <p>OpenWindows users do not typically need to use viewprint. It is automatically invoked by docviewer as needed.</p> |
| OPTIONS | <p>-P <i>printer</i> Send document to <i>printer</i></p> <p>-f <i>filename</i> Send document to <i>filename</i> (instead of printer)</p> <p>-# <i>ncopies</i> Print <i>ncopies</i> copies</p> <p>-p <i>pages</i> Specify page numbers of pages to print. <i>pages</i> can be a single page number, or a page range of the form <i>first-last</i></p> <p>-R Reverse the order of pages (last page first)</p> <p>-c <i>card-catalog</i> Specify the name of the card catalog file used to locate AnswerBooks. See ab_cardcatalog(4) for more information.</p> |
| ENVIRONMENT | <p>AB_CARDATALOG Specify the name of the card catalog file used to locate AnswerBooks. See ab_cardcatalog(4) for more information.</p> |
| SEE ALSO | docviewer (1), lp (1), ab_cardcatalog (4) |
| DIAGNOSTICS | viewprint prints an error message on failure and returns a non-zero exit status. |
| NOTES | <p>viewprint assumes the printer is PostScript-capable. In addition, most AnswerBook documents are typeset using the Palatino font, and will not print correctly on a printer that does not support that font.</p> <p>PostScript is a trademark of Adobe Systems Incorporated. AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc.</p> |

| | |
|--------------------|--|
| NAME | viewres – graphical class browser for Xt |
| SYNOPSIS | viewres [-option ...] |
| DESCRIPTION | The viewres program displays a tree showing the widget class hierarchy of the Athena Widget Set. Each node in the tree can be expanded to show the resources that the corresponding class adds (i.e. does not inherit from its parent) when a widget is created. This application allows the user to visually examine the structure and inherited resources for the Athena Widget Set. |
| OPTIONS | <p>Viewres accepts all of the standard toolkit command line options as well as the following:</p> <p>-top <i>name</i> This option specifies the name of the highest widget in the hierarchy to display. This is typically used to limit the display to a subset of the tree. The default is <i>Object</i>.</p> <p>-variable This option indicates that the widget variable names (as declared in header files) should be displayed in the nodes rather than the widget class name. This is sometimes useful to distinguish widget classes that share the same name (such as <i>Text</i>).</p> <p>-vertical This option indicates that the tree should be displayed top to bottom rather left to right.</p> |
| VIEW MENU | <p>The way in which the tree is displayed may be changed with through the entries in the View menu:</p> <p>Show Variable Names This entry causes the node labels to be set to the variable names used to declare the corresponding widget class. This operation may also be performed with the SetLabelType(variable) translation.</p> <p>Show Class Names This entry causes the node labels to be set to the class names used when specifying resources. This operation may also be performed with the SetLabelType(class) translation.</p> <p>Layout Horizontal This entry causes the tree to be laid out from left to right. This operation may also be performed with the <i>SetOrientation(West)</i> translation.</p> <p>Layout Vertical This entry causes the tree to be laid out from top to bottom. This operation may also be performed with the <i>SetOrientation(North)</i> translation.</p> <p>Show Resource Boxes This entry expands the selected nodes (see next section) to show the new widget</p> |

and constraint resources. This operation may also be performed with the *Resources(on)* translation.

Hide Resource Boxes

This entry removes the resource displays from the selected nodes (usually to conserve space). This operation may also be performed with the *Resources(off)* translation.

SELECT MENU

Resources for a single widget class can be displayed by clicking **Button2** on the corresponding node, or by adding the node to the selection list with **Button1** and using the **Show Resource Boxes** entry in the **View** menu. Since **Button1** actually toggles the selection state of a node, clicking on a selected node will cause it to be removed from the selected list.

Collections of nodes may also be selected through the various entries in the **Select** menu:

Unselect All

This entry removes all nodes from the selection list. This operation may also be performed with the *Select(nothing)* translation.

Select All

This entry adds all nodes to the selection list. This operation may also be performed with the *Select(all)* translation.

Invert All

This entry adds unselected nodes to, and removes selected nodes from, the selection list. This operation may also be performed with the *Select(invert)* translation.

Select Parent

This entry selects the immediate parents of all selected nodes. This operation may also be performed with the *Select(parent)* translation.

Select Ancestors

This entry recursively selects all parents of all selected nodes. This operation may also be performed with the *Select(ancestors)* translation.

Select Children

This entry selects the immediate children of all selected nodes. This operation may also be performed with the *Select(children)* translation.

Select Descendants

This entry recursively selects all children of all selected nodes. This operation may also be performed with the *Select(descendants)* translation.

Select Has Resources

This entry selects all nodes that add new resources (regular or constraint) to their corresponding widget classes. This operation may also be performed with the *Select(resources)* translation.

Select Shown Resource Boxes

This entry selects all nodes whose resource boxes are currently expanded (usually so that they can be closed with **Hide Resource Boxes**). This operation may

also be performed with the *Select(shown)* translation.

ACTIONS

The following application actions are provided:

Quit()

This action causes **viewres** to exit.

SetLabelType(*type*)

This action sets the node labels to display the widget *variable* or *class* names, according to the argument *type*.

SetOrientation(*direction*)

This action sets the root of the tree to be one of the following areas of the window: *West, North, East, or South*.

Select(*what*)

This action selects the indicated nodes, as described in the **VIEW MENU** section: *nothing* (unselects all nodes), *invert*, *parent*, *ancestors*, *children*, *descendants*, *resources*, *shown*.

Resources(*op*)

This action turns *on*, *off*, or *toggles* the resource boxes for the selected nodes. If invoked from within one of the nodes (through the keyboard or pointer), only that node is used.

**WIDGET
HIERARCHY**

Resources may be specified for the following widgets:

Viewres viewres

 Paned pane

 Box buttonbox

 Command quit

 MenuButton view

 SimpleMenu viewMenu

 SmeBSB layoutHorizontal

 SmeBSB layoutVertical

 SmeLine line1

 SmeBSB namesVariable

 SmeBSB namesClass

 SmeLine line2

 SmeBSB viewResources

 SmeBSB viewNoResources

 MenuButton select

 SimpleMenu selectMenu

 SmeBSB unselect

 SmeBSB selectAll

 SmeBSB selectInvert

 SmeLine line1

 SmeBSB selectParent

 SmeBSB selectAncestors

SmeBSB selectChildren
SmeBSB selectDescendants
SmeLine line2
SmeBSB selectHasResources
SmeBSB selectShownResources
Form treeform
Porthole porthole
Tree tree
Box *variable-name*
Toggle *variable-name*
List *variable-name*
Panner panner

where *variable-name* is the widget variable name of each node.

SEE ALSO X11(7), xrdp(1), listres(1), editres(1), appres(1), appropriate widget documents

COPYRIGHT Copyright 1990, Massachusetts Institute of Technology.
See X11 (7) for a full statement of rights and permissions.

AUTHOR Jim Fulton, MIT X Consortium

| | |
|--------------------|--|
| NAME | vkbd – virtual keyboard/soft function keys display manager |
| SYNOPSIS | vkbd [-nopopup] [<i>generic-tool-arguments</i>] |
| DESCRIPTION | <p>vkbd provides functionality to label the function keys specific to the individual application. When the "Function Keys" item is selected from the Workspace Utilities submenu, a panel containing the same number of function keys as your keyboard is displayed at the bottom of the screen. You can also use the keys in the Function Keys panel to activate the functions (assuming you have the Workspace property "Set Input Area" set to "Click SELECT"): just click SELECT on a key. The labels are automatically updated if you change the input area to another window. If the input area is changed to a window which does not use the function keys, the panel is updated to clear the labels.</p> <p>vkbd also provides a mechanism to switch between multiple languages by pressing and holding the LANG key. The LANG key is labeled PrSc or Print Screen by default. The LANG key can be remapped by adding the following to your \$HOME/.Xdefaults file:</p> <pre>OpenWindows.KeyboardCommand.Translate: key</pre> <p>vkbd is usually started from openwin-sys and is not intended to be used directly.</p> |
| OPTIONS | <p>-nopopup This option indicates that Function Keys popup panel is not mapped at startup time.</p> |
| FILES | \$OPENWINHOME/lib/openwin-sys |
| SEE ALSO | xview(7) <i>OPENLOOK Graphical User Interface Functional Specification</i> |

| | |
|------------------------|---|
| NAME | winsysck – check which window system protocols are available |
| SYNOPSIS | winsysck [-va] [-display <i>displaystring</i>] <i>protocol</i> [. . .] |
| DESCRIPTION | The winsysck command determines if the specified window system <i>protocol</i> is available to the user. By default, the winsysck command exits as soon as the first available <i>protocol</i> is found, although this behaviour can be modified by the -a option (see below). |
| OPTIONS | <p>The following options can be used to modify the behaviour of the winsysck command.</p> <p>-a Continue to check the availability of the specified <i>protocols</i> even after determining that one or more previously specified <i>protocols</i> are available. This is not particularly useful without the -v option (see below).</p> <p>-v Print the name of the first available <i>protocol</i> on the standard output. When combined with the -a option (see above), print the name of all available <i>protocols</i> on the standard output, separated by newlines.</p> <p>-display <i>displaystring</i> Use the display <i>displaystring</i> when trying to determine if the x11 <i>protocol</i> is available.</p> |
| KNOWN PROTOCOLS | <p>The following are known values for <i>protocol</i>.</p> <p>x11 Determines if a connection can be made using the X11 Window System protocol.</p> <p>news Determines if a connection can be made using the Network-extensible Window System protocol.</p> <p>x11news Determines if connections can be made to an X11/NeWS server. In addition to being able to establish connections using both the X11 Window System and Network-extensible Window System protocols, this requires that these two connections actually interact with the same window server.</p> <p>sunview Determines if connections can be made using the SunView Window System protocol.</p> |
| EXAMPLES | <p>Determine if an X11 Window System connection can be made:</p> <pre>example% if winsysck x11 ; then ...</pre> <p>Determine if an X11 Window System connection can be made to the display "displayhost:0.0":</p> <pre>example% if winsysck -display displayhost:0.0 x11 ; then ...</pre> <p>Print the known protocols which are available:</p> <pre>example% winsysck -v -a x11 news x11news sunview</pre> |

Print which window system should be used, given the preference for NeWS over SunView, and X11/NeWS over NeWS:

```
example% winsysck -v x11news news sunview
```

SEE ALSO

Xsun(1)

LIMITATIONS

There is no way to determine whether a SunView connection is actually connected to an X11/NeWS server with SunView compatability enabled or to a SunView server.

BUGS

None known.

DIAGNOSTICS

Exit status is 0 if any *protocols* are available, 1 if none are available, 2 for usage errors.

| | |
|--------------------|--|
| NAME | worm – draw wiggly worms |
| SYNOPSIS | worm [<i>-l length</i>] [<i>-s size</i>] [<i>-n number</i>] [<i>-d connection</i>] [<i>-g geometry</i>] [<i>-R</i>] [<i>-C</i>] [<i>-S</i>] |
| DESCRIPTION | worm draws wiggly worms. It is adapted from a concept in the December 1987 issue of Scientific American. Playing with the various parameters can create strange effects. Pressing any key in the worm window will cause them to freeze; pressing again will thaw. |
| OPTIONS | -S Screensaver. Takes over entire screen. -C Chromocolor. Worms change colors as they crawl. -R Rotate colormap. The colormap constantly changes. -n number Make <i>number</i> worms. Default is 50. -l length Worms are of length <i>length</i> . A negative value means infinite length. -size size Worms are <i>size</i> pixels wide. -display connection Connect to X server display, <i>connection</i> . -geometry geomspec Create window using <i>geomspec</i> . |
| SEE ALSO | X11(7) |
| COPYRIGHT | Copyright (c) 1988 by Sun Microsystems, Inc. David Lemke (lemke@wirehead.sun.com) Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. |

| | |
|----------------------|---|
| NAME | X11 – a portable, network-transparent window system |
| SYNOPSIS | <p>There is no X11 command per se. This manual page is adapted from the X manual page supplied with the MIT sample server and is included with OpenWindows for reference.</p> <p>The X Window System is a network transparent window system developed at MIT which runs on a wide range of computing and graphics machines.</p> <p>The X Consortium requests that the following names be used when referring to this software:</p> |
| | <pre> X X Window System X Version 11 X Window System, Version 11 X11 </pre> |
| DESCRIPTION | <p>X Window System servers run on computers with bitmap displays. The server distributes user input to and accepts output requests from various client programs through a variety of different interprocess communication channels. Although the most common case is for the client programs to be running on the same machine as the server, clients can be run transparently from other machines (including machines with different architectures and operating systems) as well.</p> <p>X supports overlapping hierarchical subwindows and text and graphics operations, on both monochrome and color displays. For a full explanation of the functions that are available, see the Xlib - C Language X Interface manual, the <i>X Window System Protocol</i> specification, the <i>X Toolkit Intrinsics - C Language Interface</i> manual, and various toolkit documents.</p> <p>Many utilities, window managers, games, toolkits, etc. are available from the user-contributed software. See your site administrator for details.</p> |
| STARTING UP | See openwin(1) for information on starting the server and an initial set of client applications. |
| DISPLAY NAMES | <p>From the user's prospective, every X server has a <i>display name</i> of the form:</p> <p style="text-align: center;"><i>hostname:displaynumber.screennumber</i></p> <p>This information is used by the application to determine how it should connect to the server and which screen it should use by default (on displays with multiple monitors):</p> <p><i>hostname</i></p> <p style="padding-left: 2em;">The <i>hostname</i> specifies the name of the machine to which the display is physically connected. If the hostname is not given, the most efficient way of communicating to a server on the same machine will be used.</p> <p><i>displaynumber</i></p> |

The phrase "display" is usually used to refer to collection of monitors that share a common keyboard and pointer (mouse, tablet, etc.). Most workstations tend to only have one keyboard, and therefore, only one display. Larger, multi-user systems, however, will frequently have several displays so that more than one person can be doing graphics work at once. To avoid confusion, each display on a machine is assigned a *display number* (beginning at 0) when the X server for that display is started. The display number must always be given in a display name.

screennumber

Some displays share a single keyboard and pointer among two or more monitors. Since each monitor has its own set of windows, each screen is assigned a *screen number* (beginning at 0) when the X server for that display is started. If the screen number is not given, then screen 0 will be used.

On POSIX systems, the default display name is stored in your DISPLAY environment variable. This variable is set automatically by the **xterm**(1) terminal emulator. However, when you log into another machine on a network, you'll need to set DISPLAY by hand to point to your display. Examples include,

```
% setenv DISPLAY myws:0
```

```
$ DISPLAY=myws:0; export DISPLAY
```

Finally, most X programs accept a command line option of **-display** *displayname* to temporarily override the contents of DISPLAY. This is most commonly used to pop windows on another person's screen or as part of a "remote shell" command to start an xterm pointing back to your display. For example,

```
% xeyes -display joesws:0 -geometry 1000x1000+0+0
```

```
% rsh big xterm -display myws:0 -ls </dev/null &
```

X servers listen for connections on a variety of different communications channels (network byte streams, shared memory, etc.). Since there can be more than one way of contacting a given server, The *hostname* part of the display name is used to determine the type of channel (also called a transport layer) to be used. The sample servers from MIT support the following types of connections:

local

The hostname part of the display name should be the empty string. For example: *:0*, *:1*, and *:0.1*. The most efficient local transport will be chosen.

TCP/IP

The hostname part of the display name should be the server machine's IP address name. Full Internet names, abbreviated names, and IP addresses are all allowed. For example: *expo.lcs.mit.edu:0*, *expo:0*, *18.30.0.212:0*, *bigmachine:1*, and *hydra:0.1*.

DECnet

The hostname part of the display name should be the server machine's

nodename followed by two colons instead of one. For example: *myws::0*, *big::1*, and *hydra::0.1*. Note that DECnet connections are not supported under Solaris x86.

ACCESS CONTROL

The sample server provides two types of access control: an authorization protocol which provides a list of “magic cookies” clients can send to request access, and a list of hosts from which connections are always accepted. **Xdm** initializes magic cookies in the server, and also places them in a file accessible to the user. Normally, the list of hosts from which connections are always accepted should be empty, so that only clients with are explicitly authorized can connect to the display. When you add entries to the host list (with **xhost(1)**), the server no longer performs any authorization on connections from those machines.

The file for authorization which both **xdm(1)** and **Xlib** use can be specified with the environment variable **XAUTHORITY**, and defaults to the file **.Xauthority** in the home directory. **Xdm** uses **\$HOME/.Xauthority** and will create it or merge in authorization records if it already exists when a user logs in.

To manage a collection of authorization files containing a collection of authorization records use **xauth(1)**. This program allows you to extract records and insert them into other files. Using this, you can send authorization to remote machines when you login. As the files are machine-independent, you can also simply copy the files or use NFS to share them. If you use several machines, and share a common home directory with NFS, then you never really have to worry about authorization files, the system should work correctly by default. Note that magic cookies transmitted “in the clear” over NFS or using **ftp(1)** or **rpc(3N)** can be “stolen” by a network eavesdropper, and as such may enable unauthorized access. In many environments this level of security is not a concern, but if it is, you need to know the exact semantics of the particular magic cookie to know if this is actually a problem.

GEOMETRY SPECIFICATIONS

One of the advantages of using window systems instead of hardwired terminals is that applications don't have to be restricted to a particular size or location on the screen. Although the layout of windows on a display is controlled by the window manager that the user is running (described below), most X programs accept a command line argument of the form **-geometry WIDTHxHEIGHT+XOFF+YOFF** (where *WIDTH*, *HEIGHT*, *XOFF*, and *YOFF* are numbers) for specifying a preferred size and location for this application's main window.

The *WIDTH* and *HEIGHT* parts of the geometry specification are usually measured in either pixels or characters, depending on the application. The *XOFF* and *YOFF* parts are measured in pixels and are used to specify the distance of the window from the left or right and top and bottom edges of the screen, respectively. Both types of offsets are measured from the indicated edge of the screen to the corresponding edge of the window. The X offset may be specified in the following ways:

+XOFF The left edge of the window is to be placed *XOFF* pixels in from the left edge of the screen (i.e. the X coordinate of the window's origin will be *XOFF*). *XOFF* may be negative, in which case the window's left edge will be off the screen.

-XOFF The right edge of the window is to be placed *XOFF* pixels in from the right edge of the screen. *XOFF* may be negative, in which case the window's right edge will be off the screen.

The Y offset has similar meanings:

+YOFF The top edge of the window is to be *YOFF* pixels below the top edge of the screen (i.e. the Y coordinate of the window's origin will be *YOFF*). *YOFF* may be negative, in which case the window's top edge will be off the screen.

-YOFF The bottom edge of the window is to be *YOFF* pixels above the bottom edge of the screen. *YOFF* may be negative, in which case the window's bottom edge will be off the screen.

Offsets must be given as pairs; in other words, in order to specify either *XOFF* or *YOFF* both must be present. Windows can be placed in the four corners of the screen using the following specifications:

+0+0 upper left hand corner.

-0+0 upper right hand corner.

-0-0 lower right hand corner.

+0-0 lower left hand corner.

In the following examples, a terminal emulator will be placed in roughly the center of the screen and a load average monitor, mailbox, and clock will be placed in the upper right hand corner:

```
xterm -fn 6x10 -geometry 80x24+30+200 &
xclock -geometry 48x48-0+0 &
xload -geometry 48x48-96+0 &
xbiff -geometry 48x48-48+0 &
```

WINDOW MANAGERS

The layout of windows on the screen is controlled by special programs called *window managers*. Although many window managers will honor geometry specifications as given, others may choose to ignore them (requiring the user to explicitly draw the window's region on the screen with the pointer, for example).

Since window managers are regular (albeit complex) client programs, a variety of different user interfaces can be built. OpenWindows comes with a window manager named **olwm(1)** which supports overlapping windows, popup menus, point-and-click or click-to-type input models, title bars, nice icons, and many other features.

FONT NAMES

Collections of characters for displaying text and symbols in X are known as *fonts*. A font typically contains images that share a common appearance and look nice together (for example, a single size, boldness, slant, and character set).

The list of font directories in which the server looks when trying to find a font is controlled by the *font path*. Although most installations will choose to have the server start up with all of the commonly used font directories, the font path can be changed at any time with the **xset(1)** program. However, it is important to remember that the directory

names are on the **server's** machine, not on the application's.

The default font path for the OpenWindows server is `$OPENWINHOME/lib/fonts`.

Font databases are created by running the **mkfontdir**(1) program in the directory containing the compiled versions of the fonts. Whenever fonts are added to a directory, **mkfontdir** should be rerun so that the server can find the new fonts. To make the server reread the font database, reset the font path with the **xset** program. For example, to add a font to a private directory, the following commands could be used:

```
% cp newfont.fb ~/myfonts
% mkfontdir ~/myfonts
% xset fp rehash
```

The **xlsfonts**(1) program can be used to list all of the fonts that are found in font databases in the current font path. Font names tend to be fairly long as they contain all of the information needed to uniquely identify individual fonts. However, the sample server supports wildcarding of font names, so the full specification

```
-adobe-courier-medium-r-normal--10-100-75-75-m-60-iso8859-1
```

could be abbreviated as:

```
-*-courier-medium-r-normal---100-***-***-***
```

Because the shell also has special meanings for `*` and `?`, wildcarded font names should be quoted:

```
% xlsfonts -fn '-*-courier-medium-r-normal---100-***-***-***'
```

If more than one font in a given directory in the font path matches a wildcarded font name, the choice of which particular font to return is left to the server. However, if fonts from more than one directory match a name, the returned font will always be from the first such directory in the font path.

COLOR NAMES

Most applications provide ways of tailoring (usually through resources or command line arguments) the colors of various elements in the text and graphics they display. Although black and white displays don't provide much of a choice, color displays frequently allow anywhere between 16 and 16 million different colors.

Colors are usually specified by their commonly-used names (for example, *red*, *white*, or *medium slate blue*). The server translates these names into appropriate screen colors using a color database that can usually be found in `$OPENWINHOME/lib/rgb.txt`. Color names are case-insensitive, meaning that *red*, *Red*, and *RED* all refer to the same color.

Many applications also accept color specifications of the following form:

```
#rgb
#rrggbb
```

```
#rrrgggbbb
#rrrrggggbbbb
```

where *r*, *g*, and *b* are hexadecimal numbers indicating how much *red*, *green*, and *blue* should be displayed (zero being none and ffff being on full). Each field in the specification must have the same number of digits (e.g., #rrgb or #gbb are not allowed). Fields that have fewer than four digits (e.g. #rgb) are padded out with zero's following each digit (e.g. #r000g000b000). The eight primary colors can be represented as:

| | |
|---------|---|
| black | #000000000000 (no color at all) |
| red | #ffff00000000 |
| green | #0000ffff0000 |
| blue | #00000000ffff |
| yellow | #ffffff0000 (full red and green, no blue) |
| magenta | #ffff0000ffff |
| cyan | #0000ffffff |
| white | #ffffffffffff (full red, green, and blue) |

Unfortunately, RGB color specifications are highly unportable since different monitors produce different shades when given the same inputs. Similarly, color names aren't portable because there is no standard naming scheme and because the color database needs to be tuned for each monitor.

Application developers should take care to make their colors tailorable.

KEYS

The X keyboard model is broken into two layers: server-specific codes (called *keycodes*) which represent the physical keys, and server-independent symbols (called *keysyms*) which represent the letters or words that appear on the keys. Two tables are kept in the server for converting keycodes to keysyms:

modifier list

Some keys (such as Shift, Control, and Caps Lock) are known as *modifier* and are used to select different symbols that are attached to a single key (such as Shift-a generates a capital A, and Control-l generates a formfeed character `^L`). The server keeps a list of keycodes corresponding to the various modifier keys. Whenever a key is pressed or released, the server generates an *event* that contains the keycode of the indicated key as well as a mask that specifies which of the modifier keys are currently pressed. Most servers set up this list to initially contain the various shift, control, and shift lock keys on the keyboard.

keymap table

Applications translate event keycodes and modifier masks into keysyms using a *keysym table* which contains one row for each keycode and one column for various modifier states. This table is initialized by the server to correspond to normal typewriter conventions, but is only used by client programs.

Although most programs deal with keysyms directly (such as those written with the X Toolkit Intrinsics), most programming libraries provide routines for converting keysyms into the appropriate type of string (such as ISO Latin-1).

OPTIONS

Most X programs attempt to use the same names for command line options and arguments. All applications written with the X Toolkit Intrinsics automatically accept the following options:

-display *display*

This option specifies the name of the X server to use.

-geometry *geometry*

This option specifies the initial size and location of the window.

-bg *color*, **-background** *color*

Either option specifies the color to use for the window background.

-bd *color*, **-bordercolor** *color*

Either option specifies the color to use for the window border.

-bw *number*, **-borderwidth** *number*

Either option specifies the width in pixels of the window border.

-fg *color*, **-foreground** *color*

Either option specifies the color to use for text or graphics.

-fn *font*, **-font** *font*

Either option specifies the font to use for displaying text.

-iconic

This option indicates that the user would prefer that the application's windows initially not be visible as if the windows had been immediately iconified by the user. Window managers may choose not to honor the application's request.

-name

This option specifies the name under which resources for the application should be found. This option is useful in shell aliases to distinguish between invocations of an application, without resorting to creating links to alter the executable file name.

-rv, **-reverse**

Either option indicates that the program should simulate reverse video if possible, often by swapping the foreground and background colors. Not all programs honor this or implement it correctly. It is usually only used on monochrome displays.

+rv

This option indicates that the program should not simulate reverse video. This is used to override any defaults since reverse video doesn't always work properly.

-selectionTimeout

This option specifies the timeout in milliseconds within which two communicating applications must respond to one another for a selection request.

-synchronous

This option indicates that requests to the X server should be sent synchronously, instead of asynchronously. Since **Xlib** normally buffers requests to the server, errors do not necessarily get reported immediately after they occur. This option

turns off the buffering so that the application can be debugged. It should never be used with a working program.

-title *string*

This option specifies the title to be used for this window. This information is sometimes used by a window manager to provide some sort of header identifying the window.

-xnlanguage *language[_territory][.codeset]*

This option specifies the language, territory, and codeset for use in resolving resource and other filenames.

-xrm *resourcestring*

This option specifies a resource name and value to override any defaults. It is also very useful for setting resources that don't have explicit command line arguments.

RESOURCES

To make the tailoring of applications to personal preferences easier, X supports several mechanisms for storing default values for program resources (e.g. background color, window title, etc.) Resources are specified as strings of the form

*appname*subname*subsubname...: value*

that are read in from various places when an application is run. By convention, the application name is the same as the program name, but with the first letter capitalized (e.g. *Bit-map* or *Emacs*) although some programs that begin with the letter "x" also capitalize the second letter for historical reasons. The precise syntax for resources is:

| | | |
|---------------|---|---|
| ResourceLine | = | Comment ResourceSpec |
| Comment | = | "!" string <empty line> |
| ResourceSpec | = | WhiteSpace ResourceName WhiteSpace ":" WhiteSpace value |
| ResourceName | = | [Binding] ComponentName {Binding ComponentName} |
| Binding | = | "," "*" |
| WhiteSpace | = | { " " "\t" } |
| ComponentName | = | { "a"-"z" "A"-"Z" "0"-"9" "_" "-" } |
| value | = | string |
| string | = | {<any character not including "\n">} |

Note that elements enclosed in curly braces ({...}) indicate zero or more occurrences of the enclosed elements

To allow values to contain arbitrary octets, the 4-character sequence `\nnn`, where `n` is a digit in the range of "0"-"7", is recognized and replaced with a single byte that contains this sequence interpreted as an octal number. For example, a value containing a NULL byte can be stored by specifying `"\000"`.

The **Xlib** routine `XGetDefault(3X)` and the resource utilities within **Xlib** and the X Toolkit Intrinsics obtain resources from the following sources:

RESOURCE_MANAGER root window property

Any global resources that should be available to clients on all machines should

be stored in the `RESOURCE_MANAGER` property on the root window using the `xrdb` program. This is frequently taken care of when the user starts up X through the display manager or `xinit`.

application-specific files

Programs that use the X Toolkit Intrinsics will also look in the directories named by the environment variable `XUSERFILESEARCHPATH` or the environment variable `XAPPLRESDIR`, plus directories in a standard place (usually under `$OPENWINHOME/lib/X11`, but this can be overridden with the `XFILESEARCHPATH` environment variable) for application-specific resources.

`-xrm` *resourcestring*

Applications that use the X Toolkit Intrinsics can have resources specified from the command line. The *resourcestring* is a single resource name and value as shown above. Note that if the string contains characters interpreted by the shell (e.g., asterisk), they must be quoted. Any number of `-xrm` arguments may be given on the command line.

Program resources are organized into groups called *classes*, so that collections of individual resources (each of which are called *instances*) can be set all at once. By convention, the instance name of a resource begins with a lowercase letter and class name with an upper case letter. Multiple word resources are concatenated with the first letter of the succeeding words capitalized. Applications written with the X Toolkit Intrinsics will have at least the following resources:

background (class **Background**)

This resource specifies the color to use for the window background.

borderWidth (class **BorderWidth**)

This resource specifies the width in pixels of the window border.

borderColor (class **BorderColor**)

This resource specifies the color to use for the window border.

Most applications using the X Toolkit Intrinsics also have the resource **foreground** (class **Foreground**), specifying the color to use for text and graphics within the window.

By combining class and instance specifications, application preferences can be set quickly and easily. Users of color displays will frequently want to set Background and Foreground classes to particular defaults. Specific color instances such as text cursors can then be overridden without having to define all of the related resources. For example,

```
bitmap*Dashed: off
XTerm*cursorColor: gold
XTerm*multiScroll: on
XTerm*jumpScroll: on
XTerm*reverseWrap: on
XTerm*curses: on
XTerm*Font: 6x10
XTerm*scrollBar: on
```

```

XTerm*scrollbar*thickness: 5
XTerm*multiClickTime: 500
XTerm*charClass: 33:48,37:48,45-47:48,64:48
XTerm*cutNewline: off
XTerm*cutToBeginningOf3ine: off
XTerm*titeInhibit: on
XTerm*ttyModes: intr ^c erase ^? kill ^u
XLoad*Background: gold
XLoad*Foreground: red
XLoad*highlight: black
XLoad*borderWidth: 0
emacs*Geometry: 80x65-0-0
emacs*Background: #5b7686
emacs*Foreground: white
emacs*Cursor: white
emacs*BorderColor: white
emacs*Font: 6x10
xmag*geometry: -0-0
xmag*borderColor: white

```

If these resources were stored in a file called **.Xresources** in your home directory, they could be added to any existing resources in the server with the following command:

```
% xrdp -merge $HOME/.Xresources
```

This is how the openwin startup script merges user-specific defaults into any site-wide defaults. All sites are encouraged to set up convenient ways of automatically loading resources. See the **Xlib** manual section *Using the Resource Manager* for more information.

EXAMPLES

The following is a collection of sample command lines for some of the more frequently used commands. For more information on a particular command, please refer to that command's manual page.

```

% xrdp -load $HOME/.Xresources
% xmodmap -e "keysym BackSpace = Delete"
% mkfontdir /usr/local/lib/otherfonts
% xset fp+ /usr/local/lib/otherfonts
% xmodmap $HOME/.keymap.km
% xsetroot -solid '#888'
% xset b 100 400 c 50 s 1800 r on
% xset q
% olwm
% xmag
% xclock -geometry 48x48-0+0 -bg blue -fg white
% xeyes -geometry 48x48-48+0
% xbiff -update 20

```

```
% xlsfonts '*helvetica*'
% xlswins -l
% xwininfo -root
% xdpinfo -display joesworkstation:0
% xhost -joesworkstation
% xrefresh
% xwd | xwud
% bitmap companylogo.bm 32x32
% xcalc -bg blue -fg magenta
% xterm -geometry 80x66-0-0 -name myxterm $*
```

DIAGNOSTICS

A wide variety of error messages are generated from various programs. Various toolkits are encouraged to provide a common mechanism for locating error text so that applications can be tailored easily. Programs written to interface directly to the **Xlib** C language library are expected to do their own error checking.

The default error handler in **Xlib** (also used by many toolkits) uses standard resources to construct diagnostic messages when errors occur. The defaults for these messages are usually stored in `$OPENWINHOME/lib/XErrorDB`. If this file is not present, error messages will consist of error codes only.

When the X Toolkit Intrinsics encounter errors converting resource strings to the appropriate internal format, no error messages are usually printed. This is convenient when it is desirable to have one set of resources across a variety of displays (e.g. color vs. monochrome, lots of fonts vs. very few, etc.), although it can pose problems for trying to determine why an application might be failing. This behavior can be overridden by the setting the *StringConversionsWarning* resource.

To force the X Toolkit Intrinsics to always print string conversion error messages, the following resource should be placed at the top of the file that gets loaded onto the `RESOURCE_MANAGER` property using the `xrdb(1)` program (frequently called *.Xresources* or *.Xres* in the user's home directory):

```
*StringConversionWarnings: on
```

To have conversion messages printed for just a particular application, the appropriate instance name can be placed before the asterisk:

```
xterm*StringConversionWarnings: on
```

SEE ALSO

appres(1), **bdfstnf(1)**, **bitmap(1)**, **mkfontdir(1)**, **makebdf(1)**, **imake(1)**, **listres(1)**, **maze(6)**, **mkfontdir(1)**, **muncher(6)**, **oclock(1)**, **olwm(1)**, **puzzle(6)**, **resize(1)**, **showsnf(1)**, **twm(1)**, **xauth(1)**, **xbiff(1)**, **xcalc(1)**, **xclipboard(1)**, **xclock(1)**, **xditview(1)**, **xdm(1)**, **xdpyinfo(1)**, **xedit(1)**, **xev(6)**, **xeyes(6)**, **xfd(1)**, **xfontsel(1)**, **xhost(1)**, **xinit(1)**, **xkill(1)**, **xload(1)**, **xlogo(1)**, **xlsatoms(1)**, **xlsclients(1)**, **xlsfonts(1)**, **xlswins(1)**, **xmag(1)**, **xman(1)**, **xmh(1)**, **xmodmap(1)**, **xpr(1)**, **xprop(1)**, **xrdb(1)**, **xrefresh(1)**, **xset(1)**, **xsetroot(1)**, **xstdcmap(1)**, **xterm(1)**, **xview(7)**, **xwd(1)**, **xwininfo(1)**, **xwud(1)**

Xlib – C Language X Interface, *X Toolkit Intrinsics – C Language Interface*, and *Using and*

*Specifying X Resources***COPYRIGHT**

The following copyright and permission notice outlines the rights and restrictions covering most parts of the core distribution of the X Window System from MIT. Other parts have additional or different copyrights and permissions; see the individual source files.

Copyright 1984, 1985, 1986, 1987, 1988, and 1989, by the Massachusetts Institute of Technology.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of MIT not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. MIT makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

TRADEMARKS

UNIX and OPEN LOOK are trademarks of AT&T. X Window System is a trademark of MIT. OpenWindows is a trademark of Sun Microsystems.

| | |
|--------------------|---|
| NAME | xauth – X authority file utility |
| SYNOPSIS | xauth [-f <i>authfile</i>] [-v] [-q] [-i] [-b] [<i>command arg...</i>] |
| DESCRIPTION | The xauth program is used to edit and display 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). Commands (described below) may be entered interactively, on the xauth command line, or in scripts. Note that this program does not contact the X server. |
| OPTIONS | <p>The following options may be used with xauth. They may be given individually (e.g. -q -i) or may be combined (e.g. -qi):</p> <p>-f <i>authfile</i> This option specifies the name of the authority file to use. By default, xauth will use the file specified by the XAUTHORITY environment variable or .Xauthority in the user's home directory.</p> <p>-q This option indicates that xauth should operate quietly and not print unsolicited status messages. This is the default if an xauth command is given on the command line or if the standard output is not directed to a terminal.</p> <p>-v This option indicates that xauth should operate verbosely and print status messages indicating the results of various operations (e.g. how many records have been read in or written out). This is the default if xauth is reading commands from its standard input and its standard output is directed to a terminal.</p> <p>-i This option indicates that xauth should ignore any authority file locks. Normally, xauth will refuse to read or edit any authority files that have been locked by other programs (usually xdm(1) or another xauth).</p> <p>-b This option indicates that xauth should attempt to break any authority file locks before proceeding and should only be used to clean up stale locks.</p> |
| COMMANDS | <p>The following commands may be used to manipulate authority files:</p> <p>add <i>displayname protocolname hexkey</i> An authorization entry for the indicated display using the given protocol and key data is added to the authorization file. The data is specified as an even-lengthed string of hexadecimal digits, each pair representing one octet. The first digit of each pair gives the most significant 4 bits of the octet and the second digit of the pair gives the least significant 4 bits. For example, a 32 character hex-key would represent a 128-bit value. A protocol name consisting of just a single period is treated as an abbreviation for <i>MIT-MAGIC-COOKIE-1</i>.</p> <p>[n]extract <i>filename displayname...</i> Authorization entries for each of the specified displays are written to the indicated file. If the <i>nextract</i> command is used, the entries are written in a numeric format suitable for non-binary transmission (such as secure electronic mail).</p> |

The extracted entries can be read back in using the *merge* and *nmerge* commands. If the filename consists of just a single dash, the entries will be written to the standard output.

[n]list [*displayname...*]

Authorization entries for each of the specified displays (or all if no displays are named) are printed on the standard output. If the *nlist* command is used, entries will be shown in the numeric format used by the *nextract* command; otherwise, they are shown in a textual format. Key data is always displayed in the hexadecimal format given in the description of the *add* command.

[n]merge [*filename...*]

Authorization entries are read from the specified files and are merged into the authorization database, superceding any matching existing entries. If the *nmerge* command is used, the numeric format given in the description of the *extract* command is used. If a filename consists of just a single dash, the standard input will be read if it hasn't been read before.

remove *displayname...*

Authorization entries matching the specified displays are removed from the authority file.

source *filename*

The specified file is treated as a script containing **xauth** commands to execute. Blank lines and lines beginning with a sharp sign (#) are ignored. A single dash may be used to indicate the standard input, if it hasn't already been read.

info Information describing the authorization file, whether or not any changes have been made, and from where **xauth** commands are being read is printed on the standard output.

exit If any modifications have been made, the authority file is written out (if allowed), and the program exits. An end of file is treated as an implicit *exit* command.

quit The program exits, ignoring any modifications. This may also be accomplished by pressing the interrupt character.

help [*string*]

A description of all commands that begin with the given string (or all commands if no string is given) is printed on the standard output.

? A short list of the valid commands is printed on the standard output.

DISPLAY NAMES

Display names for the *add*, *[n]extract*, *[n]list*, *[n]merge*, and *remove* commands use the same format as the DISPLAY environment variable and the common *-display* command line argument. Display-specific information (such as the screen number) is unnecessary and will be ignored. Same-machine connections (such as local-host sockets, shared memory, and the Internet Protocol hostname *localhost*) are referred to as *hostname/unix:displaynumber* so that local entries for different machines may be stored in one authority file.

EXAMPLE The most common use for **xauth** is to extract the entry for the current display, copy it to another machine, and merge it into the user's authority file on the remote machine:

```
example% xauth extract - $DISPLAY | rsh other xauth merge -
```

ENVIRONMENT This **xauth** program uses the following environment variables:

XAUTHORITY

to get the name of the authority file to use if the *-f* option isn't used. If this variable is not set, **xauth** will use *.Xauthority* in the user's home directory.

HOME to get the user's home directory if **XAUTHORITY** isn't defined.

BUGS Users that have unsecure networks should take care to use encrypted file transfer mechanisms to copy authorization entries between machines. Similarly, the *MIT-MAGIC-COOKIE-1* protocol is not very useful in unsecure environments. Sites that are interested in additional security may need to use encrypted authorization mechanisms such as Kerberos.

Spaces are currently not allowed in the protocol name. Quoting could be added for the truly perverse.

COPYRIGHT Copyright 1989, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR Jim Fulton, MIT X Consortium

| | |
|--------------------|---|
| NAME | xbiff – mailbox flag for X |
| SYNOPSIS | xbiff [- <i>toolkitoption</i> ...] [-option ...] |
| DESCRIPTION | <p>The xbiff program displays a little image of a mailbox. When there is no mail, the flag on the mailbox is down. When mail arrives, the flag goes up and the mailbox beeps. By default, pressing any mouse button in the image forces xbiff to remember the current size of the mail file as being the “empty” size and to lower the flag.</p> <p>This program is nothing more than a wrapper around the Athena Mailbox widget.</p> |
| OPTIONS | <p>Xbiff accepts all of the standard X Toolkit command line options along with the additional options listed below:</p> <p>-help This option indicates that a brief summary of the allowed options should be printed on the standard error.</p> <p>-update <i>seconds</i> This option specifies the frequency in seconds at which xbiff should update its display. If the mailbox is obscured and then exposed, it will be updated immediately. The default is 30 seconds.</p> <p>-file <i>filename</i> This option specifies the name of the file which should be monitored. By default, it watches <i>/var/mail/username</i>, where <i>username</i> is your login name.</p> <p>-volume <i>percentage</i> This option specifies how loud the bell should be rung when new mail comes in.</p> <p>-shape This option indicates that the mailbox window should be shaped if masks for the empty or full images are given.</p> <p>The following standard X Toolkit command line arguments are commonly used with xbiff:</p> <p>-display <i>display</i> This option specifies the X server to contact.</p> <p>-geometry <i>geometry</i> This option specifies the preferred size and position of the mailbox window. The mailbox is 48 pixels wide and 48 pixels high and will be centered in the window.</p> <p>-bg <i>color</i> This option specifies the color to use for the background of the window.</p> <p>-bd <i>color</i> This option specifies the color to use for the border of the window.</p> <p>-bw <i>number</i> This option specifies the width in pixels of the border surrounding the window.</p> <p>-fg <i>color</i> This option specifies the color to use for the foreground of the window.</p> |

-rv This option indicates that reverse video should be simulated by swapping the foreground and background colors.

-xrm *resourcestring*

This option specifies a resource string to be used. This is especially useful for setting resources that do not have separate command line options.

X DEFAULTS

The application class name is XBiff. This program uses the **Mailbox** widget in the Athena widget set. It understands all of the core resource names and classes as well as:

checkCommand (class **CheckCommand**)

Specifies a shell command to be executed to check for new mail rather than examining the size of **file**. The specified string value is used as the argument to a **system**(3) call and may therefore contain i/o redirection. An exit status of 0 indicates that new mail is waiting, 1 indicates that there has been no change in size, and 2 indicates that the mail has been cleared. By default, no shell command is provided.

file (class **File**)

Specifies the name of the file to monitor. The default is to watch */var/mail/username*, where *username* is your login name.

onceOnly

(class **Boolean**)

Specifies that the bell is only rung the first time new mail is found and is not rung again until at least one interval has passed with no mail waiting. The window will continue to indicate the presence of new mail until it has been retrieved. The default is false.

width (class **Width**)

Specifies the width of the mailbox.

height (class **Height**)

Specifies the height of the mailbox.

update (class **Interval**) Specifies the frequency in seconds at which the mail should be checked. The default is 30.

volume (class **Volume**)

Specifies how loud the bell should be rung. The default is 33 percent.

foreground

(class **Foreground**)

Specifies the color for the foreground.

reverseVideo

(class **ReverseVideo**)

Specifies that the foreground and background should be reversed.

flip (class **Flip**)

Specifies whether or not the image that is shown when mail has arrived should be inverted. The default is "true."

fullPixmap

(class **Pixmap**)

Specifies a bitmap to be shown when new mail has arrived. The default is `flagup`.

emptyPixmap

(class **Pixmap**)

Specifies a bitmap to be shown when no new mail is present. The default is `flagdown`.

shapeWindow

(class **ShapeWindow**)

Specifies whether or not the mailbox window should be shaped to the given `fullPixmapMask` and `emptyPixmapMask`. The default is `false`.

fullPixmapMask

(class **PixmapMask**)

Specifies a mask for the bitmap to be shown when new mail has arrived. The default is `none`.

emptyPixmapMask

(class **PixmapMask**)

Specifies a mask for the bitmap to be shown when no new mail is present. The default is `none`.

ACTIONS

The *Mailbox* widget provides the following actions for use in event translations:

check() This action causes the widget to check for new mail and display the flag appropriately.

unset() This action causes the widget to lower the flag until new mail comes in.

set() This action causes the widget to raise the flag until the user resets it.

The default translation is

`<ButtonPress>: unset()`

ENVIRONMENT

DISPLAY

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the `RESOURCE_MANAGER` property.

SEE ALSO

X11(7), **xrdb(1)**, **stat(2)**

BUGS

The mailbox bitmaps are ugly.

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Jim Fulton, MIT X Consortium
Additional hacks by Ralph Swick, DEC/MIT Project Athena

| | |
|--------------------|---|
| NAME | xcalc – scientific calculator for X |
| SYNOPSIS | xcalc [-stipple] [-rpn] [-toolkitoption...] |
| DESCRIPTION | xcalc is a scientific calculator desktop accessory that can emulate a TI-30 or an HP-10C. |
| OPTIONS | <p>xcalc accepts all of the standard toolkit command line options along with two additional options:</p> <p>-stipple This option indicates that the background of the calculator should be drawn using a stipple of the foreground and background colors. On monochrome displays improves the appearance.</p> <p>-rpn This option indicates that Reverse Polish Notation should be used. In this mode the calculator will look and behave like an HP-10C. Without this flag, it will emulate a TI-30.</p> |
| OPERATION | <p><i>Pointer Usage:</i> Operations may be performed with pointer button 1, or in some cases, with the keyboard. Many common calculator operations have keyboard accelerators. To quit, press pointer button 3 on the AC key of the TI calculator, or the ON key of the HP calculator.</p> <p><i>Calculator Key Usage (TI mode):</i> The numbered keys, the +/- key, and the +, -, *, /, and = keys all do exactly what you would expect them to. It should be noted that the operators obey the standard rules of precedence. Thus, entering "3+4*5=" results in "23", not "35". The parentheses can be used to override this. For example, "(1+2+3)*(4+5+6)=" results in "6*15=90".</p> <p>The entire number in the calculator display can be selected, in order to paste the result of a calculation into text.</p> <p>The action procedures associated with each function are given below. These are useful if you are interested in defining a custom calculator. The action used for all digit keys is digit(n), where <i>n</i> is the corresponding digit, 0..9.</p> <p>1/x Replaces the number in the display with its reciprocal. The corresponding action procedure is reciprocal().</p> <p>x^2 Squares the number in the display. The corresponding action procedure is square().</p> <p>SQRT Takes the square root of the number in the display. The corresponding action procedure is squareRoot().</p> <p>CE/C When pressed once, clears the number in the display without clearing the state of the machine. Allows you to re-enter a number if you make a mistake. Pressing it twice clears the state, also. The corresponding action procedure for TI mode is clear().</p> <p>AC Clears the display, the state, and the memory. Pressing it with the third pointer button turns off the calculator, in that it exits the program. The action</p> |

| | |
|----------------------|---|
| | procedure to clear the state is off() ; to quit, quit() . |
| INV | Invert function. See the individual function keys for details. The corresponding action procedure is inverse() . |
| sin | Computes the sine of the number in the display, as interpreted by the current DRG mode (see DRG, below). If inverted, it computes the arcsine. The corresponding action procedure is sine() . |
| cos | Computes the cosine, or arccosine when inverted. The corresponding action procedure is cosine() . |
| tan | Computes the tangent, or arctangent when inverted. The corresponding action procedure is tangent() . |
| DRG | Changes the DRG mode, as indicated by 'DEG', 'RAD', or 'GRAD' at the bottom of the calculator "liquid crystal" display. When in 'DEG' mode, numbers in the display are taken as being degrees. In 'RAD' mode, numbers are in radians, and in 'GRAD' mode, numbers are in grads. When inverted, the DRG key has a feature of converting degrees to radians to grads and vice-versa. Example: put the calculator into 'DEG' mode, and enter "45 INV DRG". The display should now show something along the lines of ".785398", which is 45 degrees converted to radians. The corresponding action procedure is degree() . |
| e | The constant 'e'. (2.7182818...). The corresponding action procedure is e() . |
| EE | Used for entering exponential numbers. For example, to get "-2.3E-4" you'd enter "2 . 3 +/- EE 4 +/-". The corresponding action procedure is scientific() . |
| log | Calculates the log (base 10) of the number in the display. When inverted, it raises "10.0" to the number in the display. For example, entering "3 INV log" should result in "1000". The corresponding action procedure is logarithm() . |
| ln | Calculates the log (base e) of the number in the display. When inverted, it raises "e" to the number in the display. For example, entering "e ln" should result in "1". The corresponding action procedure is naturalLog() . |
| y^x | Raises the number on the left to the power of the number on the right. For example "2 y ^x 3 =" results in "8", which is 2 ³ . For a further example, "(1+2+3) y ^x (1+2) =" equals "6 y ^x 3" which equals "216". The corresponding action procedure is power() . |
| PI | The constant 'pi'. (3.1415927....) The corresponding action procedure is pi() . |
| x! | Computes the factorial of the number in the display. The number in the display must be an integer in the range 0-500, though, depending on your math library, it might overflow long before that. The corresponding action procedure is factorial() . |
| (| Left parenthesis. The corresponding action procedure for TI calculators is leftParen() . |
|) | Right parenthesis. The corresponding action procedure for TI calculators is rightParen() . |

| | |
|------------|---|
| / | Division. The corresponding action procedure is divide() . |
| * | Multiplication. The corresponding action procedure is multiply() . |
| - | Subtraction. The corresponding action procedure is subtract() . |
| + | Addition. The corresponding action procedure is add() . |
| = | Perform calculation. The TI-specific action procedure is equal() . |
| STO | Copies the number in the display to the memory location. The corresponding action procedure is store() . |
| RCL | Copies the number from the memory location to the display. The corresponding action procedure is recall() . |
| SUM | Adds the number in the display to the number in the memory location. The corresponding action procedure is sum() . |
| EXC | Swaps the number in the display with the number in the memory location. The corresponding action procedure for the TI calculator is exchange() . |
| +/- | Negate; change sign. The corresponding action procedure is negate() . |
| . | Decimal point. The action procedure is decimal() . |

Calculator Key Usage (RPN mode): The number keys, CHS (change sign), +, -, *, /, and ENTR keys all do exactly what you would expect them to do. Many of the remaining keys are the same as in TI mode. The differences are detailed below. The action procedure for the ENTR key is **enter()**.

| | |
|-----------------------|---|
| <- | This is a backspace key that can be used if you make a mistake while entering a number. It will erase digits from the display. (See BUGS). Inverse backspace will clear the X register. The corresponding action procedure is back() . |
| ON | Clears the display, the state, and the memory. Pressing it with the third pointer button turns off the calculator, in that it exits the program. To clear state, the action procedure is off ; to quit, quit() . |
| INV | Inverts the meaning of the function keys. This would be the <i>f</i> key on an HP calculator, but <i>xcalc</i> does not display multiple legends on each key. See the individual function keys for details. |
| 10^x | Raises "10.0" to the number in the top of the stack. When inverted, it calculates the log (base 10) of the number in the display. The corresponding action procedure is tenpower() . |
| e^x | Raises "e" to the number in the top of the stack. When inverted, it calculates the log (base e) of the number in the display. The action procedure is epower() . |
| STO | Copies the number in the top of the stack to a memory location. There are 10 memory locations. The desired memory is specified by following this key with a digit key. |
| RCL | Pushes the number from the specified memory location onto the stack. |

SUM Adds the number on top of the stack to the number in the specified memory location.

x:y Exchanges the numbers in the top two stack positions, the X and Y registers. The corresponding action procedure is **XexchangeY()**.

R v Rolls the stack downward. When inverted, it rolls the stack upward. The corresponding action procedure is **roll()**.

blank These keys were used for programming functions on the HP-10C. Their functionality has not been duplicated in *xcalc*.

Finally, there are two additional action procedures: **bell()**, which rings the bell; and **selection()**, which performs a cut on the entire number in the calculator's "liquid crystal" display.

ACCELERATORS

Accelerators are shortcuts for entering commands. *xcalc* provides some sample keyboard accelerators; also users can customize accelerators. The numeric keypad accelerators provided by *xcalc* should be intuitively correct. The accelerators defined by *xcalc* on the main keyboard are given below:

| TI Key | HP Key | Keyboard Accelerator | TI Function | HP Function |
|--------|--------|----------------------|--------------|--------------|
| SQRT | SQRT | r | squareRoot() | squareRoot() |
| AC | ON | space | clear() | clear() |
| AC | <- | Delete | clear() | back() |
| AC | <- | Backspace | clear() | back() |
| AC | <- | Control-H | clear() | back() |
| AC | | Clear | clear() | |
| AC | ON | q | quit() | quit() |
| AC | ON | Control-C | quit() | quit() |
| INV | i | i | inverse() | inverse() |
| sin | s | s | sine() | sine() |
| cos | c | c | cosine() | cosine() |
| tan | t | t | tangent() | tangent() |
| DRG | DRG | d | degree() | degree() |
| e | | e | e() | |
| ln | ln | l | naturalLog() | naturalLog() |
| y^x | y^x | ^ | power() | power() |
| PI | PI | p | pi() | pi() |
| x! | x! | ! | factorial() | factorial() |
| (| | (| leftParen() | |
|) | |) | rightParen() | |
| / | / | / | divide() | divide() |
| * | * | * | multiply() | multiply() |
| - | - | - | subtract() | subtract() |

| | | | | |
|------|------|----------|-----------|--------------|
| + | + | + | add() | add() |
| = | | = | equal() | |
| 0..9 | 0..9 | 0..9 | digit() | digit() |
| . | . | . | decimal() | decimal() |
| +/- | CHS | n | negate() | negate() |
| | x:y | x | | XexchangeY() |
| | ENTR | Return | | enter() |
| | ENTR | Linefeed | | enter() |

CUSTOMIZATION

The application class name is XCalc.

xcalc has an enormous application defaults file which specifies the position, label, and function of each key on the calculator. It also gives translations to serve as keyboard accelerators. Because these resources are not specified in the source code, you can create a customized calculator by writing a private application defaults file, using the Athena Command and Form widget resources to specify the size and position of buttons, the label for each button, and the function of each button.

The foreground and background colors of each calculator key can be individually specified. For the TI calculator, a classical color resource specification might be:

```
XCalc.ti.Command.background: gray50
XCalc.ti.Command.foreground: white
```

For each of buttons 20, 25, 30, 35, and 40, specify:

```
XCalc.ti.button20.background: black
XCalc.ti.button20.foreground: white
```

For each of buttons 22, 23, 24, 27, 28, 29, 32, 33, 34, 37, 38, and 39:

```
XCalc.ti.button22.background: white
XCalc.ti.button22.foreground: black
```

**WIDGET
HIERARCHY**

In order to specify resources, it is useful to know the hierarchy of the widgets which compose *xcalc*. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

```
XCalc xcalc
  Form ti or hp (the name depends on the mode)
    Form bevel
      Form screen
        Label M
        Toggle LCD
        Label INV
        Label DEG
        Label RAD
```

| | |
|------------------------------|---|
| | Label GRAD |
| | Label P |
| | Command button1 |
| | Command button2 |
| | Command button3 |
| | <i>and so on, ...</i> |
| | Command button38 |
| | Command button39 |
| | Command button40 |
| APPLICATION RESOURCES | <p>rpn (Class Rpn) Specifies that the rpn mode should be used. The default is TI mode.</p> <p>stipple (Class Stipple) Indicates that the background should be stippled. The default is “on” for monochrome displays, and “off” for color displays.</p> <p>cursor (Class Cursor) The name of the symbol used to represent the pointer. The default is “hand2”.</p> |
| COLORS | <p>If you would like xcalc to use its ti colors, include the following in the #ifdef COLOR section of the file you read with xrdb:</p> <pre>*customization: -color</pre> <p>This will cause xcalc to pick up the colors in the app-defaults color customization file: /usr/openwin/lib/app-defaults/XCalc-color.</p> |
| SEE ALSO | X11(7) , xrdb(1) , the Athena Widget Set |
| BUGS | HP mode: A bug report claims that the sequence of keys 5, ENTER, <- should clear the display, but it doesn't. |
| COPYRIGHT | Copyright 1988, 1989, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHORS | John Bradley, University of Pennsylvania Mark Rosenstein, MIT Project Athena Donna Converse, MIT X Consortium |

| | |
|--------------------|---|
| NAME | xclipboard – X clipboard client |
| SYNOPSIS | xclipboard [<i>-toolkitoption ...</i>] [-w] [-nw] |
| DESCRIPTION | <p>The xclipboard program is used to collect and display 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. Each time CLIPBOARD is asserted by another application, xclipboard transfers the contents of that selection to a new buffer and displays it in the text window. Buffers are never automatically deleted, so you'll want to use the delete button to get rid of useless items.</p> <p>Since xclipboard uses a Text Widget to display the contents of the clipboard, text sent to the CLIPBOARD may be re-selected for use in other applications. xclipboard also responds to requests for the CLIPBOARD selection from other clients by sending the entire contents of the currently displayed buffer.</p> <p>An xclipboard window has the following buttons across the top:</p> <p><i>quit</i> When this button is pressed, xclipboard exits.</p> <p><i>delete</i> When this button is pressed, the current buffer is deleted and the next one displayed.</p> <p><i>new</i> Creates a new buffer with no contents. Useful in constructing a new CLIPBOARD selection by hand.</p> <p><i>next</i> Displays the next buffer in the list.</p> <p><i>previous</i> Displays the previous buffer.</p> |
| OPTIONS | <p>The xclipboard program accepts all of the standard X Toolkit command line options as well as the following:</p> <p>-w This option indicates that lines of text that are too long to be displayed on one line in the clipboard should wrap around to the following lines.</p> <p>-nw This option indicates that long lines of text should not wrap around. This is the default behavior.</p> |
| WIDGETS | <p>In order to specify resources, it is useful to know the hierarchy of the widgets which compose xclipboard. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.</p> <pre> XClipboard xclipboard Form form Command quit Command delete Command new Command next Command prev </pre> |

Text text

**SENDING/RETRIEVING
CLIPBOARD
CONTENTS**

Text is copied to the clipboard whenever a client asserts ownership of the **CLIPBOARD** selection. Text is copied from the clipboard whenever a client requests the contents of the **CLIPBOARD** selection. Examples of event bindings that a user may wish to include in a resource configuration file to use the clipboard are:

```
*VT100.Translations: #override \
    <Btn3Up>:                select-end(CLIPBOARD) \n\
    <Btn2Up>:                insert-selection(PRIMARY,CLIPBOARD) \n\
    <Btn2Down>:              ignore ()
```

SEE ALSO

X11(7), **xterm(1)**, individual client documentation for how to make a selection and send it to the CLIPBOARD.

ENVIRONMENT**DISPLAY**

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

FILES

/usr/openwin/lib/app-defaults/XClipboard - specifies required resources

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology
See *X11(7)* for a full statement of rights and permissions.

AUTHOR

Ralph R. Swick, DEC/MIT Project Athena
Chris D. Peterson, MIT X Consortium
Keith Packard, MIT X Consortium

| | |
|--------------------|---|
| NAME | xclock – analog/digital clock for X |
| SYNOPSIS | xclock [<i>toolkitoption...</i>] [-help] [-analog] [-digital] [-chime] [-hd color] [-hl color] [-update seconds] [-padding number] |
| DESCRIPTION | The xclock program displays the time in analog or digital form. The time is continuously updated at a frequency which may be specified by the user. This program is nothing more than a wrapper around the Athena Clock widget. |
| OPTIONS | <p>Xclock accepts all of the standard X Toolkit command line options along with the additional options listed below:</p> <p>-help This option indicates that a brief summary of the allowed options should be printed on the standard error.</p> <p>-analog This option indicates that a conventional 12 hour clock face with tick marks and hands should be used. This is the default.</p> <p>-digital This option indicates that a 24 hour digital clock should be used.</p> <p>-chime This option indicates that the clock should chime once on the half hour and twice on the hour.</p> <p>-hd color This option specifies the color of the hands on an analog clock. The default is <i>black</i>.</p> <p>-hl color This option specifies the color of the edges of the hands on an analog clock, and is only useful on color displays. The default is <i>black</i>.</p> <p>-update seconds This option specifies the frequency in seconds at which xclock should update its display. If the clock is obscured and then exposed, it will be updated immediately. A value of less than 30 seconds will enable a second hand on an analog clock. The default is 60 seconds.</p> <p>-padding number This option specifies the width in pixels of the padding between the window border and clock text or picture. The default is 10 on a digital clock and 8 on an analog clock.</p> |
| X DEFAULTS | <p>This program uses the <i>Athena Clock</i> widget. It understands all of the core resource names and classes as well as:</p> <p>width (class Width) Specifies the width of the clock. The default for analog clocks is 164 pixels; the default for digital clocks is whatever is needed to hold the clock when displayed in the chosen font.</p> <p>height (class Height) Specifies the height of the clock. The default for analog clocks is 164 pixels; the</p> |

default for digital clocks is whatever is needed to hold the clock when displayed in the chosen font.

update (class **Interval**)

Specifies the frequency in seconds at which the time should be redisplayed.

foreground (class **Foreground**)

Specifies the color for the tic marks. The default is depends on whether *reverseVideo* is specified. If *reverseVideo* is specified the default is *lwhite*, otherwise the default is *black*.

hands (class **Foreground**)

Specifies the color of the insides of the clock's hands. The default is depends on whether *reverseVideo* is specified. If *reverseVideo* is specified the default is *lwhite*, otherwise the default is *black*.

highlight (class **Foreground**)

Specifies the color used to highlight the clock's hands. The default is depends on whether *reverseVideo* is specified. If *reverseVideo* is specified the default is *lwhite*, otherwise the default is *black*.

analog (class **Boolean**)

Specifies whether or not an analog clock should be used instead of a digital one. The default is True.

chime (class **Boolean**)

Specifies whether or not a bell should be rung on the hour and half hour.

padding (class **Margin**)

Specifies the amount of internal padding in pixels to be used. The default is 8.

font (class **Font**)

Specifies the font to be used for the digital clock. Note that variable width fonts currently will not always display correctly.

WIDGETS

In order to specify resources, it is useful to know the hierarchy of the widgets which compose **xclock**. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

```
XClock xclock
    Clock clock
```

ENVIRONMENT

DISPLAY

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the **RESOURCE_MANAGER** property.

FILES /usr/openwin/lib/app-defaults/XClock - specifies required resources

SEE ALSO **X11(7)**, **xrdb(1)**, **time(1)**, Athena Clock widget

BUGS **Xclock** believes the system clock.
When in digital mode, the string should be centered automatically.

COPYRIGHT Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHORS Tony Della Fera (MIT-Athena, DEC)
Dave Mankins (MIT-Athena, BBN)
Ed Moy (UC Berkeley)

| | |
|--------------------|---|
| NAME | xcmsdb – Xlib Screen Color Characterization Data utility |
| SYNOPSIS | xcmsdb [-query] [-remove] [-color] [-format 32 16 8] [<i>filename</i>] |
| DESCRIPTION | <p>xcmsdb is used to load, query, or remove Screen Color Characterization Data stored in properties on the root window of the screen. Screen Color Characterization Data is an integral part of Xlib, necessary for proper conversion between device-independent and device-dependent color specifications. Xlib uses the XDCCC_LINEAR_RGB_MATRICES and XDCCC_LINEAR_RGB_CORRECTION properties to store color characterization data for color monitors. It uses XDCCC_GRAY_SCREENWHITEPOINT and XDCCC_GRAY_CORRECTION properties for gray scale monitors. Because Xlib allows the addition of Screen Color Characterization Function Sets, added function sets may place their Screen Color Characterization Data on other properties. This utility is unaware of these other properties, therefore, you will need to use a similar utility provided with the function set, or use the xprop(1) utility.</p> <p>The ASCII readable contents of <i>filename</i> (or the standard input if no input file is given) are appropriately transformed for storage in properties, provided the -query or -remove options are not specified.</p> |
| OPTIONS | <p>xcmsdb program accepts the following options:</p> <p>-query This option attempts to read the XDCCC properties off the screen's root window. If successful, it transforms the data into a more readable format, then sends the data to standard out.</p> <p>-remove This option attempts to remove the XDCCC properties on the screen's root window.</p> <p>-color This option sets the query and remove options to only check for the XDCCC_LINEAR_RGB_MATRICES and XDCCC_LINEAR_RGB_CORRECTION properties. If the -color option is not set then the query and remove options check for all the properties.</p> <p>-format 32 16 8 Specifies the property format (32, 16, or 8 bits per entry) for the XDCCC_LINEAR_RGB_CORRECTION property. Precision of encoded floating point values increases with the increase in bits per entry. The default is 32 bits per entry.</p> |
| SEE ALSO | xprop (1), Xlib documentation |
| ENVIRONMENT | DISPLAY to figure out which display and screen to use. |
| COPYRIGHT | Copyright 1990, Tektronix Inc. |

AUTHOR

Chuck Adams, Tektronix Inc.

| | |
|--------------------|--|
| NAME | xcolor – displays 256 colors in an X window. |
| SYNOPSIS | xcolor [-display <i>display</i>] [-geometry <i>geometry</i>] [-dump] [-nobw] [-half] [-noinst] [-iconwin] |
| DESCRIPTION | xcolor displays all 256 colors in a window. When you move the pointer into the window, installs a colormap containing a hue ramp with constant saturation and constant brightness, while preserving black and white. |
| OPTIONS | -display <i>connection</i> Connect to X server display, <i>connection</i> . -geometry <i>geomspec</i> Set window size and placement to the standard X11 geometry specification, <i>geomspec</i> . -dump Dumps the RGB values of the default colormap to stdout. -nobw Don't preserve black and white in the hue ramp. -half Create the hue ramp using only the upper half of the colormap. -noinst Don't install the hue ramp. -iconwin Use as miniature version of the main window as the icon. |
| COPYRIGHT | Copyright (c) 1989 by Sun Microsystems, Inc. Patrick J. Naughton (naughton@wind.sun.com) Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. |

| | |
|--------------------|---|
| NAME | xconsole – monitor system console messages |
| SYNOPSIS | xconsole [<i>toolkitoption...</i>] [-file <i>filename</i>] [-notify] [-stripNonprint] [-daemon] [-verbose] [-exitOnFail] |
| DESCRIPTION | The xconsole program displays messages which are usually sent to /dev/console. |
| OPTIONS | <p>Xconsole accepts all of the standard X Toolkit command line options along with the additional options listed below:</p> <p>-file <i>file-name</i> To monitor some other device, use this option to specify the device name. This does not work on regular files as they are always ready to be read from.</p> <p>-notify -nonotify When new data are received from the console and the notify option is set, the icon name of the application has " *" appended, so that it is evident even when the application is iconified. -notify is the default.</p> <p>-daemon This option causes xconsole to place itself in the background, using fork/exit.</p> <p>-verbose When set, this option directs xconsole to display an informative message in the first line of the text buffer.</p> <p>-exitOnFail When set, this option directs xconsole to exit when it is unable to redirect the console output.</p> |
| X DEFAULTS | This program uses the <i>Athena Text</i> widget, look in the <i>Athena Widget Set</i> documentation for controlling it. |
| WIDGETS | <p>In order to specify resources, it is useful to know the hierarchy of the widgets which compose xconsole. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.</p> <pre>XConsole xconsole XConsole text</pre> |
| ENVIRONMENT | <p>DISPLAY to get the default host and display number.</p> <p>XENVIRONMENT to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.</p> |

| | |
|------------------|---|
| FILES | /usr/openwin/lib/app-defaults/XConsole - specifies required resources |
| SEE ALSO | X11(7) , xrdb(1) , Athena Text widget |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Keith Packard (MIT X Consortium) |

| | |
|--------------------|---|
| NAME | xcutsel – interchange between cut buffer and selection |
| SYNOPSIS | xcutsel [<i>-toolkitoption...</i>] [-selection <i>selection</i>] [-cutbuffer <i>number</i>] |
| DESCRIPTION | <p>The xcutsel program is used to copy the current selection into a cut buffer and to make 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.</p> <p>By default, xcutsel will use the selection named PRIMARY and the cut buffer CUT_BUFFER0. Either or both of these can be overridden by command line arguments or by resources.</p> <p>An xcutsel window has the following buttons:</p> <ul style="list-style-type: none"> <i>quit</i> When this button is pressed, xcutsel exits. Any selections held by xcutsel are automatically released. <i>copy PRIMARY to 0</i> When this button is pressed, xcutsel copies the current selection into the cut buffer. <i>copy 0 to PRIMARY</i> When this button is pressed, xcutsel converts the current contents of the cut buffer into the selection. <p>The button labels reflect the selection and cutbuffer selected by command line options or through the resource database.</p> <p>When the "copy 0 to PRIMARY" button is activated, the button will remain inverted as long as xcutsel remains the owner of the selection. This serves to remind you which client owns the current selection. Note that the value of the selection remains constant; if the cutbuffer is changed, you must again activate the copy button to retrieve the new value when desired.</p> |
| OPTIONS | <p>xcutsel accepts all of the standard X Toolkit command line options as well as the following:</p> <ul style="list-style-type: none"> -selection <i>name</i> This option specifies the name of the selection to use. The default is PRIMARY. The only supported abbreviations for this option are "-select", "-sel" and "-s", as the standard toolkit option "-selectionTimeout" has a similar name. -cutbuffer <i>number</i> This option specifies the cut buffer to use. The default is cut buffer 0. |
| X DEFAULTS | <p>This program accepts all of the standard X Toolkit resource names and classes as well as:</p> <ul style="list-style-type: none"> selection (class Selection) This resource specifies the name of the selection to use. The default is PRIMARY. cutBuffer (class CutBuffer) This resource specifies the number of the cut buffer to use. The default is 0. |

| | |
|---------------------|--|
| WIDGET NAMES | <p>The following instance names may be used when user configuration of the labels in them is desired:</p> <p>sel-cut (class Command) This is the “copy SELECTION to BUFFER” button.</p> <p>cut-sel (class Command) This is the “copy BUFFER to SELECTION” button.</p> <p>quit (class Command) This is the “quit” button.</p> |
| SEE ALSO | <p>xclipboard(1), xterm(1) text widget documentation, individual client documentation for how to make a selection.</p> |
| BUGS | <p>There is no way to change the name of the selection or the number of the cut buffer while the program is running.</p> |
| COPYRIGHT | <p>Copyright 1988, Massachusetts Institute of Technology See X11(7) for a full statement of rights and permissions.</p> |
| AUTHOR | <p>Ralph R. Swick, DEC/MIT Project Athena</p> |

| | |
|--------------------|---|
| NAME | xditview – display ditroff DVI files |
| SYNOPSIS | xditview [<i>-toolkitoption ...</i>] [<i>-option ...</i>] [<i>filename</i>] |
| DESCRIPTION | The xditview program displays <i>ditroff</i> output on an X display. It uses no special metrics and automatically converts the printer coordinates into screen coordinates; using the user-specified screen resolution, rather than the actual resolution so that the appropriate fonts can be found. If “-” is given as the <i>filename</i> , xditview reads from standard input. If “ ” is the first character of <i>filename</i> , xditview forks <i>sh</i> to run the rest of the “file name” and uses the standard output of that command. |
| OPTIONS | <p>Xditview accepts all of the standard X Toolkit command line options along with the additional options listed below:</p> <p>-page <i>page-number</i> This option specifies the page number of the document to be displayed at start up time.</p> <p>-resolution <i>screen-resolution</i> This specifies the desired screen resolution to use; fonts will be opened by requesting this resolution field in the XLFD names.</p> <p>-noPolyText Some X servers incorrectly implement PolyText with multiple strings per request. This option suppresses the use of this feature in xditview.</p> <p>-backingStore <i>backing-store-type</i> Redisplay of the DVI window can take up to a second or so, this option causes the server to save the window contents so that when it is scrolled around the viewport, the window is painted from contents saved in backing store. <i>backing-store-type</i> can be one of Always, WhenMapped or NotUseful.</p> <p>The following standard X Toolkit command line arguments are commonly used with xditview:</p> <p>-bg <i>color</i> This option specifies the color to use for the background of the window. The default is <i>white</i>.</p> <p>-bd <i>color</i> This option specifies the color to use for the border of the window. The default is <i>black</i>.</p> <p>-bw <i>number</i> This option specifies the width in pixels of the border surrounding the window.</p> <p>-fg <i>color</i> This option specifies the color to use for displaying text. The default is <i>black</i>.</p> <p>-fn <i>font</i> This option specifies the font to be used for displaying widget text. The default is <i>fixed</i>.</p> <p>-rv This option indicates that reverse video should be simulated by swapping the</p> |

foreground and background colors.

-geometry *geometry*

This option specifies the preferred size and position of the window.

-display *host:display*

This option specifies the X server to contact.

-xrm *resourcestring*

This option specifies a resource string to be used.

X DEFAULTS

This program uses a *Dvi* widget. It understands all of the core resource names and classes as well as:

width (class **Width**)

Specifies the width of the window.

height (class **Height**)

Specifies the height of the window.

foreground (class **Foreground**)

Specifies the default foreground color.

font (class **Font**)

Specifies the font to be used for error messages.

FontMap (class **FontMap**)

To associate the *ditroff* fonts with appropriate X fonts, this string resource contains a set of new-line separated specifications, each of which consists of a ditroff name, some white space and an XLFD pattern with * characters in appropriate places to allow all sizes to be listed. The default fontMap is:

```
R    -*--times--medium--r--normal--*--*--*--*--*--*--iso8859-1\n\  
I    -*--times--medium--i--normal--*--*--*--*--*--*--iso8859-1\n\  
B    -*--times--bold--r--normal--*--*--*--*--*--*--iso8859-1\n\  
F    -*--times--bold--i--normal--*--*--*--*--*--*--iso8859-1\n\  
BI   -*--times--bold--i--normal--*--*--*--*--*--*--iso8859-1\n\  
C    -*--courier--medium--r--normal--*--*--*--*--*--*--iso8859-1\n\  
CO   -*--courier--medium--o--normal--*--*--*--*--*--*--iso8859-1\n\  
CB   -*--courier--bold--r--normal--*--*--*--*--*--*--iso8859-1\n\  
CF   -*--courier--bold--o--normal--*--*--*--*--*--*--iso8859-1\n\  
H    -*--helvetica--medium--r--normal--*--*--*--*--*--*--iso8859-1\n\  
HO   -*--helvetica--medium--o--normal--*--*--*--*--*--*--iso8859-1\n\  
HB   -*--helvetica--bold--r--normal--*--*--*--*--*--*--iso8859-1\n\  
HF   -*--helvetica--bold--o--normal--*--*--*--*--*--*--iso8859-1\n\  
N    -*--new century schoolbook--medium--r--normal--*--*--*--*--*--*--iso8859-1\n\  
NI   -*--new century schoolbook--medium--i--normal--*--*--*--*--*--*--iso8859-1\n\  
NB   -*--new century schoolbook--bold--r--normal--*--*--*--*--*--*--iso8859-1\n\  
NF   -*--new century schoolbook--bold--i--normal--*--*--*--*--*--*--iso8859-1\n\  
A    -*--charter--medium--r--normal--*--*--*--*--*--*--iso8859-1\n\  
AI   -*--charter--medium--i--normal--*--*--*--*--*--*--iso8859-1\n\  

```

```

AB    --charter-bold-r-normal--*-*-*-*-*iso8859-1\n\
AF    --charter-bold-i-normal--*-*-*-*-*iso8859-1\n\
S     --symbol-medium-r-normal--*-*-*-*-*adobe-fontspecific\n\
S2    --symbol-medium-r-normal--*-*-*-*-*adobe-fontspecific\n

```

**USING
XDITVIEW WITH
DITROFF**

You can use any DVI file with **xditview**, although DVI files which use the fonts appropriate to the fontMap will look more accurate on the screen. On servers which support scaled fonts, all requested font sizes will be accurately reflected on the screen; for servers which do not support scaled **xditview** will use the closest font from the same family.

SEE ALSO

X11(7), **xrdb(1)**, *X Logical Font Description Conventions*

ORIGIN

Portions of this program originated in *xtroff* which was derived from *suntroff*.

COPYRIGHT

Copyright 1989, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHORS

Keith Packard (MIT X Consortium)
Richard L. Hyde (Purdue)
David Slattengren (Berkeley)
Malcom Slaney (Schlumberger Palo Alto Research)
Mark Moraes (University of Toronto)

| | | | | | | | | | |
|------------------------------|---|-------------------------|--------------------------------------|------------------------------|----------------------------|---------------------------|--|-------------------------|--------------------------------------|
| NAME | xdm – X Display Manager with support for XDMCP | | | | | | | | |
| SYNOPSIS | xdm [-config <i>configuration_file</i>] [-nodaemon] [-debug <i>debug_level</i>] [-error <i>error_log_file</i>] [-resources <i>resource_file</i>] [-server <i>server_entry</i>] [-session <i>session_program</i>] | | | | | | | | |
| DESCRIPTION | <p>Xdm manages a collection of X displays, which may be on the local host or remote servers. The design of xdm was guided by the needs of X terminals as well as the X Consortium standard XDMCP, the <i>X Display Manager Control Protocol</i>. Xdm provides services similar to those provided by init(1M), getty(1M), and login(1) on character terminals: prompting for login name and password, authenticating the user, and running a “session.”</p> <p>A “session” is defined by the lifetime of a particular process; in the traditional character-based terminal world, it is the user’s login shell. In the xdm context, it is an arbitrary session manager. This is because in a windowing environment, a user’s login shell process does not necessarily have any terminal-like interface with which to connect. When a real session manager is not available, a window manager or terminal emulator is typically used as the “session manager,” meaning that termination of this process terminates the user’s session.</p> <p>When the session is terminated, xdm resets the X server and (optionally) restarts the whole process.</p> <p>When xdm receives an Indirect query via XDMCP, it can run a chooser process to perform an XDMCP BroadcastQuery (or an XDMCP Query to specified hosts) on behalf of the display and offer a menu of possible hosts that offer XDMCP display management. This feature is useful with X terminals that do not offer a host menu themselves.</p> <p>Because xdm provides the first interface that users will see, it is designed to be simple to use and easy to customize to the needs of a particular site. Xdm has many options, most of which have reasonable defaults. Browse through the various sections of this manual, picking and choosing the things you want to change. Pay particular attention to the Session Program section, which will describe how to set up the style of session desired.</p> | | | | | | | | |
| TYPICAL USAGE | <p>Actually, xdm is designed to operate in such a wide variety of environments that <i>typical</i> is probably a misnomer.</p> <p>First, the xdm configuration file should be set up. Make a directory (usually <i>/usr/openwin/lib/xdm</i>) to contain all of the relevant files. Here is a reasonable configuration file, which could be named <i>xdm-config</i>:</p> <table border="0" style="margin-left: 40px;"> <tr> <td>DisplayManager.servers:</td> <td><i>/usr/openwin/lib/xdm/Xservers</i></td> </tr> <tr> <td>DisplayManager.errorLogFile:</td> <td><i>/usr/tmp/xdm-errors</i></td> </tr> <tr> <td>DisplayManager*resources:</td> <td><i>/usr/openwin/lib/xdm/Xresources</i></td> </tr> <tr> <td>DisplayManager*startup:</td> <td><i>/usr/openwin/lib/xdm/Xstartup</i></td> </tr> </table> | DisplayManager.servers: | <i>/usr/openwin/lib/xdm/Xservers</i> | DisplayManager.errorLogFile: | <i>/usr/tmp/xdm-errors</i> | DisplayManager*resources: | <i>/usr/openwin/lib/xdm/Xresources</i> | DisplayManager*startup: | <i>/usr/openwin/lib/xdm/Xstartup</i> |
| DisplayManager.servers: | <i>/usr/openwin/lib/xdm/Xservers</i> | | | | | | | | |
| DisplayManager.errorLogFile: | <i>/usr/tmp/xdm-errors</i> | | | | | | | | |
| DisplayManager*resources: | <i>/usr/openwin/lib/xdm/Xresources</i> | | | | | | | | |
| DisplayManager*startup: | <i>/usr/openwin/lib/xdm/Xstartup</i> | | | | | | | | |

```

DisplayManager*session:      /usr/openwin/lib/xdm/Xsession
DisplayManager.pidFile:     /usr/tmp/xdm-pid
DisplayManager._0.authorize: true
DisplayManager*authorize:   false

```

Note that this file simply contains references to other files. Note also that some of the resources are specified with “*” separating the components. These resources can be made unique for each different display, by replacing the “*” with the display-name, but normally this is not very useful. See the **Resources** section for a complete discussion.

The first file, */usr/openwin/lib/xdm/Xservers*, contains the list of displays to manage that are not using XDMCP. Most workstations have only one display, numbered 0, so the file will look something like this:

```
:0 Local local /usr/openwin/bin/X :0
```

This will keep */usr/openwin/bin/X* running on this display and manage a continuous cycle of sessions.

The file */usr/tmp/xdm-errors* will contain error messages from **xdm** and anything output to stderr by *Xsetup*, *Xstartup*, *Xsession* or *Xreset*. When you have trouble getting **xdm** working, check this file to see if **xdm** has any clues to the trouble.

The next configuration entry, */usr/openwin/lib/xdm/Xresources*, is loaded onto the display as a resource database using **xrdb**(1). As the authentication widget reads this database before starting up, it usually contains parameters for that widget:

```

xlogin*login.translations: #override\
    <Key>F1: set-session-argument(failsafe) finish-field()\n\
    <Key>Return: set-session-argument() finish-field()
xlogin*borderWidth: 3
#ifdef COLOR
xlogin*greetColor: CadetBlue
xlogin*failColor: red
#endif

```

Please note the translations entry; it specifies a few new translations for the widget which allow users to escape from the default session (and avoid troubles that may occur in it). Note that if **#override** is not specified, the default translations are removed and replaced by the new value, not a very useful result as some of the default translations are quite useful (such as “<Key>: insert-char ()” which responds to normal typing).

The *Xstartup* file shown here simply prevents login while the file */etc/nologin* exists. As there is no provision for displaying any messages here (there isn’t any core X client which displays files), the user will probably be baffled by this behavior. Thus this is not a complete example, but simply a demonstration of the available functionality.

Here is a sample *Xstartup* script:

```
#!/bin/sh
#
# Xstartup
#
# This program is run as root after the user is verified
#
if [ -f /etc/nologin ]; then
    exit 1
fi
exit 0
```

The most interesting script is *Xsession*. This version recognizes the special “failsafe” mode, specified in the translations in the *Xresources* file above, to provide an escape from the ordinary session:

```
#!/bin/sh
#
# Xsession
#
# This is the program that is run as the client
# for the display manager. This example is
# quite friendly as it attempts to run a per-user
# .xsession file instead of forcing a particular
# session layout
#

case $# in
1)
    case $1 in
failsafe)
        exec xterm -geometry 80x24-0-0 -ls
        ;;
    esac
esac

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

if [ -f $startup ]; then
    exec $startup
    exec /bin/sh $startup
else
    if [ ! -f $resources ]; then
        resources=$HOME/.Xdefaults
```

```

fi
if [ -f $resources ]; then
    xrdp -load $resources
fi
twm &
exec xterm -geometry 80x24+10+10 -ls
fi

```

OPTIONS

All of these options, except **-config**, specify values which can also be specified in the configuration file as resources.

-config *configuration_file*

Names the configuration file, which specifies resources to control the behavior of **xdm**. */usr/openwin/lib/xdm/xdm-config* is the default.

-nodaemon

Specifies “false” as the value for the **DisplayManager.daemonMode** resource. This suppresses the normal daemon behavior, which is for **xdm** to close all file descriptors, disassociate itself from the controlling terminal, and put itself in the background when it first starts up.

-debug *debug_level*

Specifies the numeric value for the **DisplayManager.debugLevel** resource. A non-zero value causes **xdm** to print lots of debugging statements to the terminal; it also disables the **DisplayManager.daemonMode** resource, forcing **xdm** to run synchronously. To interpret these debugging messages, a copy of the source code for **xdm** is almost a necessity. No attempt has been made to rationalize or standardize the output.

-error *error_log_file*

Specifies the value for the **DisplayManager.errorLogFile** resource. This file contains errors from **xdm** as well as anything written to stderr by the various scripts and programs run during the progress of the session.

-resources *resource_file*

Specifies the value for the **DisplayManager*resources** resource. This file is loaded using **xrdb(1)** to specify configuration parameters for the authentication widget.

-server *server_entry*

Specifies the value for the **DisplayManager.servers** resource. See the section **Server Specification** for a description of this resource.

-udpPort *port_number*

Specifies the value for the **DisplayManager.requestPort** resource. This sets the port-number which **xdm** will monitor for XDMCP requests. As XDMCP uses the registered well-known UDP port 177, this resource should not be changed except for debugging.

-session *session_program*

Specifies the value for the **DisplayManager*session** resource. This indicates the

program to run as the session after the user has logged in.

-xrm resource_specification

Allows an arbitrary resource to be specified, as in most X Toolkit applications.

RESOURCES

At many stages the actions of **xdm** can be controlled through the use of its configuration file, which is in the X resource format. Some resources modify the behavior of **xdm** on all displays, while others modify its behavior on a single display. Where actions relate to a specific display, the display name is inserted into the resource name between “Display-Manager” and the final resource name segment. For example, **DisplayManager.expo_0.startup** is the name of the resource which defines the startup shell file on the “expo:0” display. Because the resource manager uses colons to separate the name of the resource from its value and dots to separate resource name parts, **xdm** substitutes underscores for both dots and colons when generating the resource name.

DisplayManager.servers

This resource either specifies a file name full of server entries, one per line (if the value starts with a slash), or a single server entry. See the section **Server Specification** for the details.

DisplayManager.requestPort

This indicates the UDP port number which **xdm** uses to listen for incoming XDMCP requests. Unless you need to debug the system, leave this with its default value of 177.

DisplayManager.errorLogFile

Error output is normally directed at the system console. To redirect it, set this resource to a file name. A method to send these messages to *syslog* should be developed for systems which support it; however, the wide variety of interfaces precludes any system-independent implementation. This file also contains any output directed to stderr by the *Xsetup*, *Xstartup*, *Xsession* and *Xreset* files, so it will contain descriptions of problems in those scripts as well.

DisplayManager.debugLevel

If the integer value of this resource is greater than zero, reams of debugging information will be printed. It also disables daemon mode, which would redirect the information into the bit-bucket, and allows non-root users to run **xdm**, which would normally not be useful.

DisplayManager.daemonMode

Normally, **xdm** attempts to make itself into a daemon process unassociated with any terminal. This is accomplished by forking and leaving the parent process to exit, then closing file descriptors and releasing the controlling terminal. In some environments this is not desired (in particular, when debugging). Setting this resource to “false” will disable this feature.

DisplayManager.pidFile

The filename specified will be created to contain an ASCII representation of the process-id of the main **xdm** process. **Xdm** also uses file locking on this file to attempt to eliminate multiple daemons running on the same machine, which

would cause quite a bit of havoc.

DisplayManager.lockPidFile

This is the resource which controls whether **xdm** uses file locking to keep multiple display managers from running amok. On System V, this uses the **lockf(3C)** library call, while on BSD it uses **flock(3B)**

DisplayManager.remoteAuthDir

This names a directory in which **xdm** stores authorization files while initializing the session. The default value is */usr/openwin/lib/xdm*.

DisplayManager.autoRescan

This boolean controls whether **xdm** rescans the configuration, servers, access control and authentication keys files after a session terminates and the files have changed. By default it is "true." You can force **xdm** to reread these files by sending a SIGHUP to the main process.

DisplayManager.removeDomainname

When computing the display name for XDMCP clients, the name resolver will typically create a fully qualified host name for the terminal. As this is sometimes confusing, **xdm** will remove the domain name portion of the host name if it is the same as the domain name of the local host when this variable is set. By default the value is "true."

DisplayManager.keyFile

XDM-AUTHENTICATION-1 style XDMCP authentication requires that a private key be shared between **xdm** and the terminal. This resource specifies the file containing those values. Each entry in the file consists of a display name and the shared key. By default, **xdm** does not include support for XDM-AUTHENTICATION-1, as it requires DES which is not generally distributable because of United States export restrictions.

DisplayManager.accessFile

To prevent unauthorized XDMCP service and to allow forwarding of XDMCP IndirectQuery requests, this file contains a database of hostnames which are either allowed direct access to this machine, or have a list of hosts to which queries should be forwarded to. The format of this file is described in the section **XDMCP Access Control**.

DisplayManager.DISPLAY.resources

This resource specifies the name of the file to be loaded by **xrdb(3)** as the resource database onto the root window of screen 0 of the display. The *Xsetup* program, the Login widget, and *chooser* will use the resources set in this file. This resource data base is loaded just before the authentication procedure is started, so it can control the appearance of the login window. See the section **Authentication Widget**, which describes the various resources that are appropriate to place in this file. There is no default value for this resource, but */usr/openwin/lib/xdm/Xresources* is the conventional name.

DisplayManager.DISPLAY.chooser

Specifies the program run to offer a host menu for Indirect queries redirected to

the special host name CHOOSER. */usr/openwin/lib/xdm/chooser* is the default. See the sections **XDMCP Access Control** and **Chooser**.

DisplayManager.DISPLAY.xrdb

Specifies the program used to load the resources. By default, **xdm** uses */usr/openwin/bin/xrdb*.

DisplayManager.DISPLAY.cpp

This specifies the name of the C preprocessor which is used by **xrdb(1)**.

DisplayManager.DISPLAY.setup

This specifies a program which is run (as root) before offering the Login window. This may be used to change the appearance of the screen around the Login window or to put up other windows (e.g., you may want to run *xconsole* here). By default, no program is run. The conventional name for a file used here is *Xsetup*. See the section **Setup Program**.

DisplayManager.DISPLAY.startup

This specifies a program which is run (as root) after the authentication process succeeds. By default, no program is run. The conventional name for a file used here is *Xstartup*. See the section **Startup Program**.

DisplayManager.DISPLAY.session

This specifies the session to be executed (not running as root). By default, */usr/openwin/bin/xterm* is run. The conventional name is *Xsession*. See the section **Session Program**.

DisplayManager.DISPLAY.reset

This specifies a program which is run (as root) after the session terminates. Again, by default no program is run. The conventional name is *Xreset*. See the section **Reset Program**.

DisplayManager.DISPLAY.openDelay**DisplayManager.DISPLAY.openRepeat****DisplayManager.DISPLAY.openTimeout****DisplayManager.DISPLAY.startAttempts**

These numeric resources control the behavior of **xdm** when attempting to open intransigent servers. **openDelay** is the length of the pause (in seconds) between successive attempts, **openRepeat** is the number of attempts to make, **openTimeout** is the amount of time to wait while actually attempting the open (i.e., the maximum time spent in the *connect(2)* system call) and **startAttempts** is the number of times this entire process is done before giving up on the server. After **openRepeat** attempts have been made, or if **openTimeout** seconds elapse in any particular attempt, **xdm** terminates and restarts the server, attempting to connect again. This process is repeated **startAttempts** times, at which point the display is declared dead and disabled. Although this behavior may seem arbitrary, it has been empirically developed and works quite well on most systems. The default values are 5 for **openDelay**, 5 for **openRepeat**, 30 for **openTimeout** and 4 for **startAttempts**.

DisplayManager.DISPLAY.pingInterval**DisplayManager.DISPLAY.pingTimeout**

To discover when remote displays disappear, **xdm** occasionally pings them, using an X connection and *XSync* calls. **pingInterval** specifies the time (in minutes) between each ping attempt, **pingTimeout** specifies the maximum amount of time (in minutes) to wait for the terminal to respond to the request. If the terminal does not respond, the session is declared dead and terminated. By default, both are set to 5 minutes. If you frequently use X terminals which can become isolated from the managing host, you may wish to increase this value. The only worry is that sessions will continue to exist after the terminal has been accidentally disabled. **xdm** will not ping local displays. Although it would seem harmless, it is unpleasant when the workstation session is terminated as a result of the server hanging for NFS service and not responding to the ping.

DisplayManager.DISPLAY.terminateServer

This boolean resource specifies whether the X server should be terminated when a session terminates (instead of resetting it). This option can be used when the server tends to grow without bound over time, in order to limit the amount of time the server is run. The default value is "false."

DisplayManager.DISPLAY.userPath

Xdm sets the PATH environment variable for the session to this value. It should be a colon separated list of directories; see **sh(1)** for a full description. `"/bin:/usr/bin:/usr/openwin/bin:/usr/ucb"` is a common setting. The default value can be specified at build time in the X system configuration file with `DefaultUserPath`;

DisplayManager.DISPLAY.systemPath

Xdm sets the PATH environment variable for the startup and reset scripts to the value of this resource. The default for this resource is specified at build time by the `DefaultSystemPath` entry in the system configuration file; `"/etc:/bin:/usr/bin:/usr/openwin/bin:/usr/ucb"` is a common choice. Note the absence of "." from this entry. This is a good practice to follow for root; it avoids many common Trojan Horse system penetration schemes.

DisplayManager.DISPLAY.systemShell

Xdm sets the SHELL environment variable for the startup and reset scripts to the value of this resource. It is `/bin/sh` by default.

DisplayManager.DISPLAY.failSafeClient

If the default session fails to execute, **xdm** will fall back to this program. This program is executed with no arguments, but executes using the same environment variables as the session would have had (see the section **Session Program**). By default, `/usr/openwin/bin/xterm` is used.

DisplayManager.DISPLAY.grabServer**DisplayManager.DISPLAY.grabTimeout**

To improve security, **xdm** grabs the server and keyboard while reading the login name and password. The `grabServer` resource specifies if the server should be

held for the duration of the name/password reading. When “false,” the server is ungrabbed after the keyboard grab succeeds, otherwise the server is grabbed until just before the session begins. The default is “false.” The **grabTimeout** resource specifies the maximum time **xdm** will wait for the grab to succeed. The grab may fail if some other client has the server grabbed, or possibly if the network latencies are very high. This resource has a default value of 3 seconds; you should be cautious when raising it, as a user can be spoofed by a look-alike window on the display. If the grab fails, **xdm** kills and restarts the server (if possible) and the session.

DisplayManager.DISPLAY.authorize

DisplayManager.DISPLAY.authName

authorize is a boolean resource which controls whether **xdm** generates and uses authorization for the local server connections. If authorization is used, **authName** specifies the type to use. Currently, **xdm** supports only MIT-MAGIC-COOKIE-1 authorization. XDM-AUTHORIZATION-1 could be supported as well, but DES is not generally distributable. XDMCP connections specify which authorization types are supported dynamically, so **authName** is ignored in this case. When **authorize** is set for a display and authorization is not available, the user is informed by having a different message displayed in the login widget. By default, **authorize** is “true”; **authName** is “MIT-MAGIC-COOKIE-1.”

DisplayManager.DISPLAY.authFile

This file is used to communicate the authorization data from **xdm** to the server, using the **-auth** server command line option. It should be kept in a directory which is not world-writable as it could easily be removed, disabling the authorization mechanism in the server.

DisplayManager.DISPLAY.resetForAuth

The original implementation of authorization in the sample server reread the authorization file at server reset time, instead of when checking the initial connection. As **xdm** generates the authorization information just before connecting to the display, an old server would not get up-to-date authorization information. This resource causes **xdm** to send SIGHUP to the server after setting up the file, causing an additional server reset to occur, during which time the new authorization information will be read. The default is “false,” which will work for all MIT servers.

DisplayManager.DISPLAY.userAuthDir

When **xdm** is unable to write to the usual user authorization file (**\$HOME/.Xauthority**), it creates a unique file name in this directory and points the environment variable **XAUTHORITY** at the created file. It uses **/tmp** by default.

**XDMCP ACCESS
CONTROL**

The database file specified by the **DisplayManager.accessFile** provides information which **xdm** uses to control access from displays requesting XDMCP service. This file contains three types of entries: entries which control the response to Direct and Broadcast queries, entries which control the response to Indirect queries, and macro definitions.

The format of the Direct entries is simple, either a host name or a pattern, which is distinguished from a host name by the inclusion of one or more meta characters ('*' matches any sequence of 0 or more characters, and '?' matches any single character) which are compared against the host name of the display device. If the entry is a host name, all comparisons are done using network addresses, so any name which converts to the correct network address may be used. For patterns, only canonical host names are used in the comparison, so ensure that you do not attempt to match aliases. Preceding either a host name or a pattern with a '!' character causes hosts which match that entry to be excluded.

An Indirect entry also contains a host name or pattern, but follows it with a list of host names or macros to which indirect queries should be sent.

A macro definition contains a macro name and a list of host names and other macros that the macro expands to. To distinguish macros from hostnames, macro names start with a '%' character. Macros may be nested.

Indirect entries may also specify to have **xdm** run *chooser* to offer a menu of hosts to connect to. See the section **Chooser**.

When checking access for a particular display host, each entry is scanned in turn and the first matching entry determines the response. Direct and Broadcast entries are ignored when scanning for an Indirect entry and vice-versa.

Blank lines are ignored, '#' is treated as a comment delimiter causing the rest of that line to be ignored, and 'newline' causes the newline to be ignored, allowing indirect host lists to span multiple lines.

Here is an example Xaccess file:

```
#
# Xaccess - XDMCP access control file
#

#
# Direct/Broadcast query entries
#

!xtra.lcs.mit.edu          # disallow direct/broadcast service for xtra
bambi.ogi.edu            # allow access from this particular display
*.lcs.mit.edu             # allow access from any display in LCS

#
# Indirect query entries
#

%HOSTS                    expo.lcs.mit.edu xenon.lcs.mit.edu \
                           excess.lcs.mit.edu kanga.lcs.mit.edu

extract.lcs.mit.edu       xenon.lcs.mit.edu           #force extract to contact xenon
```

```
l1tra.lcs.mit.edu          dummy          #disallow indirect access
*.lcs.mit.edu             %HOSTS        #all others get to choose
```

CHOOSER

For X terminals that do not offer a host menu for use with Broadcast or Indirect queries, the *chooser* program can do this for them. In the *Xaccess* file, specify "CHOOSER" as the first entry in the Indirect host list. *Chooser* will send a Query request to each of the remaining host names in the list and offer a menu of all the hosts that respond.

The list may consist of the word "BROADCAST," in which case *chooser* will send a Broadcast instead, again offering a menu of all hosts that respond. Note that on some operating systems, UDP packets cannot be broadcast, so this feature will not work.

Example *Xaccess* file using *chooser*:

```
extract.lcs.mit.edu       CHOOSER %HOSTS      #offer a menu of these hosts
l1tra.lcs.mit.edu        CHOOSER BROADCAST  #offer a menu of all hosts
```

The program to use for **chooser** is specified by the **DisplayManager.DISPLAY.chooser** resource. Resources for this program can be put into the file named by **DisplayManager.DISPLAY.resources**.

SERVER SPECIFICATION

The resource **DisplayManager.servers** gives a server specification or, if the values starts with a slash (/), the name of a file containing server specifications, one per line.

Each specification indicates a display which should constantly be managed and which is not using XDMCP. Each consists of at least three parts: a display name, a display class, a display type, and (for local servers) a command line to start the server. A typical entry for local display number 0 would be:

```
:0 local /usr/openwin/bin/X :0
```

The display types are:

```
local          local display: x11m must run the server
foreign        remote display: x11m opens an X connection to a running server
```

The display name must be something that can be passed in the **-display** option to an X program. This string is used to generate the display-specific resource names, so be careful to match the names (e.g. use ":0 local /usr/openwin/bin/X :0" instead of "localhost:0 local /usr/openwin/bin/X :0" if your other resources are specified as "DisplayManager._0.session"). The display class portion is also used in the display-specific resources, as the class of the resource. This is useful if you have a large collection of similar displays (like a corral of X terminals) and would like to set resources for groups of them. When using XDMCP, the display is required to specify the display class, so the manual for your particular X terminal should document the display class string for your device. If it doesn't, you can run **x11m** in debug mode and look at the resource strings which it generates for that device, which will include the class string.

**SETUP
PROGRAM**

The *Xsetup* file is run after the server is reset, but before the Login window is offered. The file is typically a shell script. It is run as root, so should be careful about security. This is the place to change the root background or bring up other windows that should appear on the screen along with the Login widget. Note that since **xdm** grabs the keyboard, any other windows will not be able to receive keyboard input. They will be able to interact with the mouse, however; beware of potential security holes here. If **DisplayManager.DISPLAY.grabServer** is set, *Xsetup* will not be able to connect to the display at all. Resources for this program can be put into the file named by **DisplayManager.DISPLAY.resources**.

**AUTHENTICATION
WIDGET**

The authentication widget reads a name/password pair from the keyboard. Nearly every imaginable parameter can be controlled with a resource. Resources for this widget should be put into the file named by **DisplayManager.DISPLAY.resources**. All of these have reasonable default values, so it is not necessary to specify any of them.

xlogin.Login.width, xlogin.Login.height, xlogin.Login.x, xlogin.Login.y

The geometry of the Login widget is normally computed automatically. If you wish to position it elsewhere, specify each of these resources.

xlogin.Login.foreground

The color used to display the typed-in user name.

xlogin.Login.font

The font used to display the typed-in user name.

xlogin.Login.greeting

A string which identifies this window. The default is "X Window System."

xlogin.Login.unsecureGreeting

When X authorization is requested in the configuration file for this display and none is in use, this greeting replaces the standard greeting. The default is "This is an unsecure session"

xlogin.Login.greetFont

The font used to display the greeting.

xlogin.Login.greetColor

The color used to display the greeting.

xlogin.Login.namePrompt

The string displayed to prompt for a user name. **Xrdb** strips trailing white space from resource values, so to add spaces at the end of the prompt (usually a nice thing), add spaces escaped with backslashes. The default is "Login: "

xlogin.Login.passwdPrompt

The string displayed to prompt for a password. The default is "Password: "

xlogin.Login.promptFont

The font used to display both prompts.

xlogin.Login.promptColor

The color used to display both prompts.

xlogin.Login.fail

A message which is displayed when the authentication fails. The default is "Login incorrect"

xlogin.Login.failFont

The font used to display the failure message.

xlogin.Login.failColor

The color used to display the failure message.

xlogin.Login.failTimeout

The number of seconds that the failure message is displayed. The default is 30.

xlogin.Login.translations

This specifies the translations used for the login widget. Refer to the X Toolkit documentation for a complete discussion on translations. The default translation table is:

| | |
|-----------------|---------------------------------|
| Ctrl<Key>H: | delete-previous-character() \n\ |
| Ctrl<Key>D: | delete-character() \n\ |
| Ctrl<Key>B: | move-backward-character() \n\ |
| Ctrl<Key>F: | move-forward-character() \n\ |
| Ctrl<Key>A: | move-to-begining() \n\ |
| Ctrl<Key>E: | move-to-end() \n\ |
| Ctrl<Key>K: | erase-to-end-of-line() \n\ |
| Ctrl<Key>U: | erase-line() \n\ |
| Ctrl<Key>X: | erase-line() \n\ |
| Ctrl<Key>C: | restart-session() \n\ |
| Ctrl<Key>\\: | abort-session() \n\ |
| <Key>BackSpace: | delete-previous-character() \n\ |
| <Key>Delete: | delete-previous-character() \n\ |
| <Key>Return: | finish-field() \n\ |
| <Key>: | insert-char() \ |

The actions which are supported by the widget are:

delete-previous-character

Erases the character before the cursor.

delete-character

Erases the character after the cursor.

move-backward-character

Moves the cursor backward.

move-forward-character

Moves the cursor forward.

move-to-begining

(Apologies about the spelling error.) Moves the cursor to the beginning of the editable text.

move-to-end

Moves the cursor to the end of the editable text.

erase-to-end-of-line

Erases all text after the cursor.

erase-line

Erases the entire text.

finish-field

If the cursor is in the name field, proceeds to the password field; if the cursor is in the password field, checks the current name/password pair. If the name/password pair is valid, **xdm** starts the session. Otherwise the failure message is displayed and the user is prompted again.

abort-session

Terminates and restarts the server.

abort-display

Terminates the server, disabling it. This is a rash action and is not accessible in the default configuration. It can be used to stop *xdm* when shutting the system down or when using *xdmshell*.

restart-session

Resets the X server and starts a new session. This can be used when the resources have been changed and you want to test them or when the screen has been overwritten with system messages.

insert-char

Inserts the character typed.

set-session-argument

Specifies a single word argument which is passed to the session at startup. See the sections **Session Program** and **Typical Usage**.

allow-all-access

Disables access control in the server. This can be used when the *.Xauthority* file cannot be created by **xdm**. Be very careful using this; it might be better to disconnect the machine from the network before doing this.

STARTUP PROGRAM

The *Xstartup* file is typically a shell script. It is run as root and should be very careful about security. This is the place to put commands which add entries to */etc/utmp*, mount users' home directories from file servers, display the message of the day, or abort the session if logins are not allowed. Various environment variables are set for the use of this script:

| | |
|------------|--|
| DISPLAY | the associated display name |
| HOME | the initial working directory of the user |
| USER | the user name |
| PATH | the value of DisplayManager.DISPLAY.systemPath |
| SHELL | the value of DisplayManager.DISPLAY.systemShell |
| XAUTHORITY | may be set to an authority file |

No arguments are passed to the script. **Xdm** waits until this script exits before starting the user session. If the exit value of this script is non-zero, **x11** discontinues the session and starts another authentication cycle.

SESSION PROGRAM

The *Xsession* program is the command which is run as the user's session. It is run with the permissions of the authorized user, and has several environment variables specified:

| | |
|------------|---|
| DISPLAY | the associated display name |
| HOME | the initial working directory of the user |
| USER | the user name |
| PATH | the value of <code>DisplayManager.DISPLAY.userPath</code> |
| SHELL | the user's default shell (from <i>getpwnam</i>) |
| XAUTHORITY | may be set to a non-standard authority file |

At most installations, *Xsession* should look in \$HOME for a file *.xsession*, which contains commands that each user would like to use as a session. *Xsession* should also implement a system default session if no user-specified session exists. See the section **Typical Usage**.

An argument may be passed to this program from the authentication widget using the 'set-session-argument' action. This can be used to select different styles of session. One good use of this feature is to allow the user to escape from the ordinary session when it fails. This allows users to repair their own *.xsession* if it fails, without requiring administrative intervention. The section **Typical Usage** demonstrates this feature.

RESET PROGRAM

Symmetrical with *Xstartup*, the *Xreset* script is run after the user session has terminated. Run as root, it should contain commands that undo the effects of commands in *Xstartup*, removing entries from */etc/utmp* or unmounting directories from file servers. The environment variables that were passed to *Xstartup* are also passed to *Xreset*.

CONTROLLING THE SERVER

X11 controls local servers using POSIX signals. SIGHUP is expected to reset the server, closing all client connections and performing other cleanup duties. SIGTERM is expected to terminate the server. If these signals do not perform the expected actions, **x11** will not perform properly.

To control remote terminals not using XDMCP, **x11** searches the window hierarchy on the display and uses the protocol request KillClient in an attempt to clean up the terminal for the next session. This may not actually kill all of the clients, as only those which have created windows will be noticed. XDMCP provides a more sure mechanism; when **x11** closes its initial connection, the session is over and the terminal is required to close all other connections.

CONTROLLING XDM

X11 responds to two signals: SIGHUP and SIGTERM. When sent a SIGHUP, **x11** rereads the configuration file, the access control file, and the servers file. For the servers file, it notices if entries have been added or removed. If a new entry has been added, **x11** starts a session on the associated display. Entries which have been removed are disabled immediately, meaning that any session in progress will be terminated without notice and no new session will be started.

When sent a SIGTERM, **xdm** terminates all sessions in progress and exits. This can be used when shutting down the system.

Xdm attempts to mark its various sub-processes for **ps(1)** by editing the command line argument list in place. Because **xdm** can't allocate additional space for this task, it is useful to start **xdm** with a reasonably long command line (using the full path name should be enough). Each process which is servicing a display is marked *-display*.

OTHER POSSIBILITIES

You can use **xdm** to run a single session at a time, using the 4.3 **init** options or other suitable daemon by specifying the server on the command line:

```
xdm -server ":0 SUN-3/60CG4 local /usr/bin/X :0"
```

Or, you might have a file server and a collection of X terminals. The configuration for this is identical to the sample above, except the *Xservers* file would look like

```
extol:0 VISUAL-19 foreign
exalt:0 NCD-19 foreign
explode:0 NCR-TOWERVIEW3000 foreign
```

This directs **xdm** to manage sessions on all three of these terminals. See the section **Controlling Xdm** for a description of using signals to enable and disable these terminals in a manner reminiscent of **init(1M)**.

LIMITATIONS

One thing that **xdm** isn't very good at doing is coexisting with other window systems. To use multiple window systems on the same hardware, you'll probably be more interested in **xinit**.

FILES

```
/usr/openwin/lib/xdm/xdm-config
    the default configuration file
/usr/openwin/lib/xdm/Xaccess
    the default access file, listing authorized displays
/usr/openwin/lib/xdm/Xservers
    the default server file, listing non-XDMCP servers to manage
$(HOME)/.Xauthority user authorization file where xdm stores keys for clients to read
/usr/openwin/lib/xdm/chooser
    the default chooser
/usr/openwin/bin/xrdb the default resource database loader
/usr/openwin/bin/X   the default server
/usr/openwin/bin/xterm the default session program and failsafe client
/usr/openwin/lib/xdm/A<host>-<suffix>
    the default place for authorization files
```

SEE ALSO X11(7), xinit(1), xauth(1), and XDMCP

COPYRIGHT Copyright 1988, Massachusetts Institute of Technology.
See X11(7) for a full statement of rights and permissions.

AUTHOR Keith Packard, MIT X Consortium

| | |
|--------------------|--|
| NAME | xdpr – dump an X window directly to a printer |
| SYNOPSIS | xdpr [<i>filename</i>] [-display <i>host:display</i>] [-P <i>printer</i>] [-device <i>printer_device</i>] [option...] |
| DESCRIPTION | <p>xdpr uses the commands xwd(1), xpr(1), and lp(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. Xdpr by default will print the largest possible representation of the window on the output page.</p> <p>The options for xdpr are the same as those for xpr, xwd, and lp. The most commonly-used options are described below; see the manual pages for these commands for more detailed descriptions of the many options available.</p> <p><i>filename</i> Specifies a file containing a window dump (created by xwd) to be printed instead of selecting an X window.</p> <p>-P printer Specifies a printer to send the output to. If a printer name is not specified here, xdpr (really, lp) will send your output to the printer specified by the <i>PRINTER</i> environment variable. Be sure that type of the printer matches the type specified with the -device option.</p> <p>-display host:display[.screen] Normally, xdpr gets the host and display number to use from the environment variable "DISPLAY". One can, however, specify them explicitly; see X11(7).</p> <p>-device printer-device Specifies the device type of the printer. Available printer devices are "ln03" for the DEC LN03, "pp" for the IBM 3812 PagePrinter, and "ps" for any postscript printer (e.g. DEC LN03R or LPS40). The default is "ln03".</p> <p>-help This option displays the list of options known to xdpr.</p> <p>Any other arguments will be passed to the xwd(1), xpr(1), and lp(1) commands as appropriate for each.</p> |
| SEE ALSO | xwd (1), xpr (1), lp (1), xwud (1) |
| ENVIRONMENT | DISPLAY - for which display to use by default. PRINTER - for which printer to use by default. |
| COPYRIGHT | Copyright 1985, 1988, Massachusetts Institute of Technology. See X11 (7) for a full statement of rights and permissions. |
| AUTHOR | Paul Boutin, MIT Project Athena Michael R. Gretzinger, MIT Project Athena Jim Gettys, MIT Project Athena |

| | |
|--------------------|---|
| NAME | xdpyinfo – display information utility for X |
| SYNOPSIS | xdpyinfo [-display <i>displayname</i>] |
| DESCRIPTION | Xdpyinfo is a utility for displaying information about an X server. It is used to examine 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. |
| EXAMPLE | <p>The following shows a sample produced by xdpyinfo when connected to display that supports an 8 plane screen and a 1 plane screen.</p> <pre> name of display: :0.0 version number: 11.0 vendor string: MIT X Consortium vendor release number: 4 maximum request size: 16384 longwords (65536 bytes) motion buffer size: 0 bitmap unit, bit order, padding: 32, MSBFirst, 32 image byte order: MSBFirst number of supported pixmap formats: 2 supported pixmap formats: depth 1, bits_per_pixel 1, scanline_pad 32 depth 8, bits_per_pixel 8, scanline_pad 32 keycode range: minimum 8, maximum 129 focus: PointerRoot number of extensions: 4 SHAPE MIT-SHM Multi-Buffering MIT-SUNDRY-NONSTANDARD default screen number: 0 number of screens: 2 screen #0: dimensions: 1152x900 pixels (325x254 millimeters) resolution: 90x90 dots per inch depths (2): 1, 8 root window id: 0x8006e depth of root window: 8 planes number of colormaps: minimum 1, maximum 1 default colormap: 0x8006b default number of colormap cells: 256 preallocated pixels: black 1, white 0 options: backing-store YES, save-unders YES </pre> |

```
current input event mask: 0xd0801d
KeyPressMask      ButtonPressMask      ButtonReleaseMask
EnterWindowMask   ExposureMask      SubstructureRedirectMask
PropertyChangeMask ColormapChangeMask
number of visuals: 6
default visual id: 0x80065
visual:
  visual id: 0x80065
  class: PseudoColor
  depth: 8 planes
  size of colormap: 256 entries
  red, green, blue masks: 0x0, 0x0, 0x0
  significant bits in color specification: 8 bits
visual:
  visual id: 0x80066
  class: DirectColor
  depth: 8 planes
  size of colormap: 8 entries
  red, green, blue masks: 0x7, 0x38, 0xc0
  significant bits in color specification: 8 bits
visual:
  visual id: 0x80067
  class: GrayScale
  depth: 8 planes
  size of colormap: 256 entries
  red, green, blue masks: 0x0, 0x0, 0x0
  significant bits in color specification: 8 bits
visual:
  visual id: 0x80068
  class: StaticGray
  depth: 8 planes
  size of colormap: 256 entries
  red, green, blue masks: 0x0, 0x0, 0x0
  significant bits in color specification: 8 bits
visual:
  visual id: 0x80069
  class: StaticColor
  depth: 8 planes
  size of colormap: 256 entries
  red, green, blue masks: 0x7, 0x38, 0xc0
  significant bits in color specification: 8 bits
visual:
  visual id: 0x8006a
  class: TrueColor
  depth: 8 planes
```

```

size of colormap: 8 entries
red, green, blue masks: 0x7, 0x38, 0xc0
significant bits in color specification: 8 bits
number of mono multibuffer types: 6
visual id, max buffers, depth: 0x80065, 0, 8
visual id, max buffers, depth: 0x80066, 0, 8
visual id, max buffers, depth: 0x80067, 0, 8
visual id, max buffers, depth: 0x80068, 0, 8
visual id, max buffers, depth: 0x80069, 0, 8
visual id, max buffers, depth: 0x8006a, 0, 8
number of stereo multibuffer types: 0

screen #1:
dimensions: 1152x900 pixels (325x254 millimeters)
resolution: 90x90 dots per inch
depths (1): 1
root window id: 0x80070
depth of root window: 1 plane
number of colormaps: minimum 1, maximum 1
default colormap: 0x8006c
default number of colormap cells: 2
preallocated pixels: black 1, white 0
options: backing-store YES, save-unders YES
current input event mask: 0xd0801d
KeyPressMask      ButtonPressMask      ButtonReleaseMask
EnterWindowMask   ExposureMask      SubstructureRedirectMask
PropertyChangeMask  ColormapChangeMask
number of visuals: 1
default visual id: 0x80064
visual:
visual id: 0x80064
class: StaticGray
depth: 1 plane
size of colormap: 2 entries
red, green, blue masks: 0x0, 0x0, 0x0
significant bits in color specification: 1 bits
number of mono multibuffer types: 1
visual id, max buffers, depth: 0x80064, 0, 1
number of stereo multibuffer types: 0

```

ENVIRONMENT**DISPLAY**

To get the default host, display number, and screen.

SEE ALSO | **X11(7)**, **xwininfo(1)**, **xprop(1)**, **xrdb(1)**

COPYRIGHT | Copyright 1988, 1989, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR | Jim Fulton, MIT X Consortium

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------------|--|------------------|---|----------------|--|------------------|--|-------------|---|--------|-----------------------|--------|-------------|--------|-------------|--------|--------------------------|--------|-------------------|--------|----------------|--------|----------------|--------|---------------|--------|---------------------------|--------|--------------------------|--------|--------------------|--------|----------------------|--------|---------------------|--------|------------------|--------|----------------|--------|-----------|--------|---------|--------|----------------------|--------|-----------|--------|--------------------|
| NAME | xedit – simple text editor for X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SYNOPSIS | xedit [<i>-toolkitoption ...</i>] [<i>filename</i>] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DESCRIPTION | <p>Xedit provides a window consisting of the following four areas:</p> <table border="0"> <tr> <td style="padding-right: 20px;">Commands Section</td> <td>A set of commands that allow you to exit xedit, save the file, or load a new file into the edit window.</td> </tr> <tr> <td>Message Window</td> <td>Displays xedit messages. In addition, this window can be used as a scratch pad.</td> </tr> <tr> <td>Filename Display</td> <td>Displays the name of the file currently being edited, and whether this file is <i>Read-Write</i> or <i>Read Only</i>.</td> </tr> <tr> <td>Edit Window</td> <td>Displays the text of the file that you are editing or creating.</td> </tr> </table> | Commands Section | A set of commands that allow you to exit xedit , save the file, or load a new file into the edit window. | Message Window | Displays xedit messages. In addition, this window can be used as a scratch pad. | Filename Display | Displays the name of the file currently being edited, and whether this file is <i>Read-Write</i> or <i>Read Only</i> . | Edit Window | Displays the text of the file that you are editing or creating. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Commands Section | A set of commands that allow you to exit xedit , save the file, or load a new file into the edit window. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Message Window | Displays xedit messages. In addition, this window can be used as a scratch pad. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Filename Display | Displays the name of the file currently being edited, and whether this file is <i>Read-Write</i> or <i>Read Only</i> . | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Edit Window | Displays the text of the file that you are editing or creating. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OPTIONS | <p>Xedit accepts all of the standard X Toolkit command line options (see X11(7)). The order of the command line options is not important.</p> <p><i>filename</i> Specifies the file that is to be loaded during start-up. This is the file which will be edited. If a file is not specified, xedit lets you load a file or create a new file after it has started up.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| EDITING | <p>The Athena Text widget is used for the three sections of this application that allow text input. The characters typed will go to the Text widget that the pointer cursor is currently over. If the pointer cursor is not over a text widget then the keypresses will have no effect on the application. This is also true for the special key sequences that popup dialog widgets, so typing Control-S in the filename widget will enable searching in that widget, not the edit widget.</p> <p>Both the message window and the edit window will create a scrollbar if the text to display is too large to fit in that window. Horizontal scrolling is not allowed by default, but can be turned on through the Text widget's resources, see <i>Athena Widget Set</i> for the exact resource definition.</p> <p>The following keystroke combinations are defined:</p> <table border="0"> <tr> <td>Ctrl-a</td> <td>Beginning Of Line</td> <td>Meta-b</td> <td>Backward Word</td> </tr> <tr> <td>Ctrl-b</td> <td>Backward Character</td> <td>Meta-f</td> <td>Forward Word</td> </tr> <tr> <td>Ctrl-d</td> <td>Delete Next Character</td> <td>Meta-i</td> <td>Insert File</td> </tr> <tr> <td>Ctrl-e</td> <td>End Of Line</td> <td>Meta-k</td> <td>Kill To End Of Paragraph</td> </tr> <tr> <td>Ctrl-f</td> <td>Forward Character</td> <td>Meta-q</td> <td>Form Paragraph</td> </tr> <tr> <td>Ctrl-g</td> <td>Multiply Reset</td> <td>Meta-v</td> <td>Previous Page</td> </tr> <tr> <td>Ctrl-h</td> <td>Delete Previous Character</td> <td>Meta-y</td> <td>Insert Current Selection</td> </tr> <tr> <td>Ctrl-j</td> <td>Newline And Indent</td> <td>Meta-z</td> <td>Scroll One Line Down</td> </tr> <tr> <td>Ctrl-k</td> <td>Kill To End Of Line</td> <td>Meta-d</td> <td>Delete Next Word</td> </tr> <tr> <td>Ctrl-l</td> <td>Redraw Display</td> <td>Meta-D</td> <td>Kill Word</td> </tr> <tr> <td>Ctrl-m</td> <td>Newline</td> <td>Meta-h</td> <td>Delete Previous Word</td> </tr> <tr> <td>Ctrl-n</td> <td>Next Line</td> <td>Meta-H</td> <td>Backward Kill Word</td> </tr> </table> | Ctrl-a | Beginning Of Line | Meta-b | Backward Word | Ctrl-b | Backward Character | Meta-f | Forward Word | Ctrl-d | Delete Next Character | Meta-i | Insert File | Ctrl-e | End Of Line | Meta-k | Kill To End Of Paragraph | Ctrl-f | Forward Character | Meta-q | Form Paragraph | Ctrl-g | Multiply Reset | Meta-v | Previous Page | Ctrl-h | Delete Previous Character | Meta-y | Insert Current Selection | Ctrl-j | Newline And Indent | Meta-z | Scroll One Line Down | Ctrl-k | Kill To End Of Line | Meta-d | Delete Next Word | Ctrl-l | Redraw Display | Meta-D | Kill Word | Ctrl-m | Newline | Meta-h | Delete Previous Word | Ctrl-n | Next Line | Meta-H | Backward Kill Word |
| Ctrl-a | Beginning Of Line | Meta-b | Backward Word | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-b | Backward Character | Meta-f | Forward Word | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-d | Delete Next Character | Meta-i | Insert File | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-e | End Of Line | Meta-k | Kill To End Of Paragraph | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-f | Forward Character | Meta-q | Form Paragraph | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-g | Multiply Reset | Meta-v | Previous Page | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-h | Delete Previous Character | Meta-y | Insert Current Selection | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-j | Newline And Indent | Meta-z | Scroll One Line Down | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-k | Kill To End Of Line | Meta-d | Delete Next Word | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-l | Redraw Display | Meta-D | Kill Word | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-m | Newline | Meta-h | Delete Previous Word | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ctrl-n | Next Line | Meta-H | Backward Kill Word | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | |
|--------|-------------------------|----------------------|----------------------|
| Ctrl-o | Newline And Backup | Meta-< | Beginning Of File |
| Ctrl-p | Previous Line | Meta-> | End Of File |
| Ctrl-r | Search/Replace Backward | Meta-] | Forward Paragraph |
| Ctrl-s | Search/Replace Forward | Meta-[| Backward Paragraph |
| Ctrl-t | Transpose Characters | | |
| Ctrl-u | Multiply by 4 | Meta-Delete | Delete Previous Word |
| Ctrl-v | Next Page | Meta-Shift Delete | Kill Previous Word |
| Ctrl-w | Kill Selection | Meta-Backspace | Delete Previous Word |
| Ctrl-y | Unkill | Meta-Shift Backspace | Kill Previous Word |
| Ctrl-z | Scroll One Line Up | | |

In addition, the pointer may be used to cut and paste text:

| | |
|-----------------|----------------------------------|
| Button 1 Down | Start Selection |
| Button 1 Motion | Adjust Selection |
| Button 1 Up | End Selection (cut) |
| Button 2 Down | Insert Current Selection (paste) |
| Button 3 Down | Extend Current Selection |
| Button 3 Motion | Adjust Selection |
| Button 3 Up | End Selection (cut) |

COMMANDS

| | |
|------|--|
| Quit | Quits the current editing session. If any changes have not been saved, xedit displays a warning message, allowing the user to save the file. |
| Save | If file backups are enabled (see RESOURCES, below) xedit stores a copy of the original, unedited file in <prefix> <i>file</i> <suffix>, then overwrites the <i>file</i> with the contents of the edit window. The filename is retrieved from the Text widget directly to the right of the <i>Load</i> button. |
| Load | Loads the file named in the text widget immediately to the right of the this button and displays it in the Edit window. If the currently displayed file has been modified a warning message will ask the user to save the changes, or press <i>Load</i> again. |

RESOURCES

For **xedit** the available resources are:

enableBackups (Class **EnableBackups**)

Specifies that, when edits made to an existing file are saved, **xedit** is to copy the original version of that file to <prefix>*file*<suffix> before it saves the changes. The default value for this resource is "off," stating that no backups should be created.

backupNamePrefix (Class **BackupNamePrefix**)

Specifies a string that is to be prepended to the backup filename. The default is that no string shall be prepended.

backupNameSuffix (Class **BackupNameSuffix**)

Specifies a string that is to be appended to the backup filename. The default is to use “.BAK” as the suffix.

WIDGETS

In order to specify resources, it is useful to know the hierarchy of the widgets which compose **xedit**. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

```
Xedit xedit
  Paned paned
    Paned buttons
      Command quit
      Command save
      Command load
      Text filename
    Label bc_label
    Text messageWindow
    Label labelWindow
    Text editWindow
```

ENVIRONMENT

DISPLAY to get the default host and display number.
XENVIRONMENT to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

FILES

/usr/openwin/lib/app-defaults/Xedit specifies required resources

SEE ALSO

X11(7), **xrdb(1)**, *Athena Widget Set*

RESTRICTIONS

There is no *undo* function.

COPYRIGHT

Copyright 1988, Digital Equipment Corporation.
 Copyright 1989, Massachusetts Institute of Technology.
 See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Chris D. Peterson, MIT X Consortium

| | |
|--------------------|---|
| NAME | xepdf – display an Encapsulated PostScript file |
| SYNOPSIS | xepdf [-display <i>name</i>] [-mag <i>n</i>] <i>filename</i> |
| DESCRIPTION | xepdf is a Display PostScript program that displays an Encapsulated PostScript file. |
| OPTIONS | -display <i>name</i> specifies the display on which to open a connection to the Display PostScript system. If no display is specified, the DISPLAY environment variable is used. -mag <i>n</i> scales the image by a factor of <i>n</i> . The scale factor may be either an integer or a floating-point number. |
| DIAGNOSTICS | Error messages are printed to standard output. |
| AUTHOR | Adobe Systems Incorporated |
| NOTES | PostScript and Display PostScript are trademarks of Adobe Systems Incorporated which may be registered in certain jurisdictions. Copyright (c) 1990-1991 Adobe Systems Incorporated. All rights reserved. |

| | |
|--------------------|--|
| NAME | xev – print contents of X events |
| SYNOPSIS | xev [-display <i>displayname</i>] [-geometry <i>geom</i>] |
| DESCRIPTION | Xev creates a window and then asks the X server to send it notices called <i>events</i> 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. |
| OPTIONS | -display <i>display</i> This option specifies the X server to contact. -geometry <i>geom</i> This option specifies the size and/or location of the window. |
| SEE ALSO | X11(7) , xwininfo(1) , xdpyinfo(1) , Xlib Programmers Manual, X Protocol Specification |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium |

| | |
|--------------------|---|
| NAME | xeyes – Eyes follow your pointer |
| SYNOPSIS | xeyes [-option ...] |
| DESCRIPTION | xeyes displays a pair of eyes which follow your pointer as it is moved around the display. |
| OPTIONS | <p>-fg <i>foreground color</i> choose a different color for the pupil of the eyes.</p> <p>-bg <i>background color</i> choose a different color for the background.</p> <p>-outline <i>outline color</i> choose a different color for the outline of the eyes.</p> <p>-center <i>center color</i> choose a different color for the center of the eyes.</p> <p>-backing WhenMapped Always NotUseful selects an appropriate level of backing store.</p> <p>-geometry <i>geometry</i> define the initial window geometry; see X11(7).</p> <p>-display <i>display</i> specify the display to use; see X11(7).</p> <p>-bd <i>border color</i> choose a different color for the window border.</p> <p>-bw <i>border width</i> choose a different width for the window border.</p> <p>-shape uses the SHAPE extension to shape the window.</p> |
| SEE ALSO | X Toolkit documentation |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Keith Packard, MIT X Consortium |

| | |
|--------------------|---|
| NAME | xfd – display all the characters in an X font |
| SYNOPSIS | xfd [<i>-toolkitoptions...</i>] [-fn <i>fontname</i>] [-box] [-center] [-start <i>number</i>] [-bc <i>color</i>] |
| DESCRIPTION | <p>The xfd utility 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. The first character displayed at the top left will be character number 0 unless the -start option has been supplied in which case the character with the number given in the -start option will be used.</p> <p>The characters are displayed in a grid of boxes, each large enough to hold any single character in the font. Each character glyph is drawn using the PolyText16 request (used by the Xlib routine XDrawString16). If the -box option is given, a rectangle will be drawn around each character, showing where an ImageText16 request (used by the Xlib routine XDrawImageString16) would cause background color to be displayed.</p> <p>The origin of each glyph is normally set so that the character is drawn in the upper left hand corner of the grid cell. However, if a glyph has a negative left bearing or an unusually large ascent, descent, or right bearing (as is the case with <i>cursor</i> font), some character may not appear in their own grid cells. The -center option may be used to force all glyphs to be centered in their respective cells.</p> <p>All the characters in the font may not fit in the window at once. To see the next page of glyphs, press the <i>Next</i> button at the top of the window. To see the previous page, press <i>Prev</i>. To exit xfd, press <i>Quit</i>.</p> <p>Individual character metrics (index, width, bearings, ascent and descent) can be displayed at the top of the window by pressing on the desired character.</p> <p>The font name displayed at the top of the window is the full name of the font, as determined by the server. See xlsfonts(1) for ways to generate lists of fonts, as well as more detailed summaries of their metrics and properties.</p> |
| OPTIONS | <p>xfd accepts all of the standard toolkit command line options along with the additional options listed below:</p> <p>-fn <i>font</i> This option specifies the font to be displayed.</p> <p>-box This option indicates that a box should be displayed outlining the area that would be filled with background color by an ImageText request.</p> <p>-center This option indicates that each glyph should be centered in its grid.</p> <p>-start <i>number</i> This option specifies the glyph index of the upper left hand corner of the grid. This is used to view characters at arbitrary locations in the font. The default is 0.</p> <p>-bc <i>color</i> This option specifies the color to be used if ImageText boxes are drawn.</p> |

| | |
|------------------|---|
| SEE ALSO | X11(7) , xlsfonts(1) , xrdb(1) , xfontsel(1) |
| BUGS | The program should skip over pages full of non-existent characters. |
| COPYRIGHT | Copyright 1989, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium; previous program of the same name by Mark Lillibridge, MIT Project Athena. |

| | |
|---------------------|--|
| NAME | xfontsel – point & click interface for selecting X11 font names |
| SYNOPSIS | xfontsel [<i>-toolkitoption...</i>] [-pattern <i>fontname</i>] [-print] [-sample <i>text</i>] [-sample16 <i>text16</i>] [-noscaled] |
| DESCRIPTION | <p>The xfontsel application provides a simple way to display the fonts known to your X server, examine samples of each, and retrieve the X Logical Font Description ("XLFD") full name for a font.</p> <p>If -pattern is not specified, all fonts with XLFD 14-part names will be selectable. To work with only a subset of the fonts, specify -pattern followed by a partially or fully qualified font name; e.g., "-pattern *medium*" will select that subset of fonts which contain the string "medium" somewhere in their font name. Be careful about escaping wildcard characters in your shell.</p> <p>If -print is specified on the command line the selected font specifier will be written to standard output when the <i>quit</i> button is activated. Regardless of whether or not -print was specified, the font specifier may be made the PRIMARY (text) selection by activating the <i>select</i> button.</p> <p>The -sample option specifies the sample text to be used to display the selected font if the font is linearly indexed, overriding the default.</p> <p>The -sample16 option specifies the sample text to be used to display the selected font if the font is matrix encoded, overriding the default.</p> <p>The -noscaled option disables the ability to select scaled fonts at arbitrary pixel or point sizes. This makes it clear which bitmap sizes are advertised by the server, and can avoid an accidental and sometimes prolonged wait for a font to be scaled.</p> |
| INTERACTIONS | <p>Clicking any pointer button in one of the XLFD field names will pop up a menu of the currently-known possibilities for that field. If previous choices of other fields were made, only values for fonts which matched the previously selected fields will be selectable; to make other values selectable, you must deselect some other field(s) by choosing the "*" entry in that field. Unselectable values may be omitted from the menu entirely as a configuration option; see the ShowUnselectable resource, below. Whenever any change is made to a field value, xfontsel will assert ownership of the PRIMARY_FONT selection. Other applications (see, e.g., xterm(1)) may then retrieve the selected font specification.</p> <p>Scalable fonts come back from the server with zero for the pixel size, point size, and average width fields. Selecting a font name with a zero in these positions results in an implementation-dependent size. Any pixel or point size can be selected to scale the font to a particular size. Any average width can be selected to anamorphically scale the font (although you may find this challenging given the size of the average width menu).</p> <p>Clicking the left pointer button in the <i>select</i> widget will cause the currently selected font name to become the PRIMARY text selection as well as the PRIMARY_FONT selection. This then allows you to paste the string into other applications. The select button remains highlighted to remind you of this fact, and de-highlights when some other application takes the PRIMARY selection away. The <i>select</i> widget is a toggle; pressing it when</p> |

it is highlighted will cause **xfontsel** to release the selection ownership and de-highlight the widget. Activating the *select* widget twice is the only way to cause **xfontsel** to release the PRIMARY_FONT selection.

RESOURCES

The application class is **XFontSel**. Most of the user-interface is configured in the app-defaults file; if this file is missing a warning message will be printed to standard output and the resulting window will be nearly incomprehensible.

Most of the significant parts of the widget hierarchy are documented in the app-defaults file (normally /usr/openwin/lib/app-defaults/XFontSel).

Application specific resources:

cursor (class **Cursor**)

Specifies the cursor for the application window.

pattern (class **Pattern**)

Specifies the font name pattern for selecting a subset of available fonts.

Equivalent to the **-pattern** option. Most useful patterns will contain at least one field delimiter; e.g. **"*-m-"** for monospaced fonts.

pixelSizeList (class **PixelSizeList**)

Specifies a list of pixel sizes to add to the pixel size menu, so that scalable fonts can be selected at those pixel sizes. The default pixelSizeList contains 7, 30, 40, 50, and 60.

pointSizeList (class **PointSizeList**)

Specifies a list of point sizes (in units of tenths of points) to add to the point size menu, so that scalable fonts can be selected at those point sizes. The default pointSizeList contains 250, 300, 350, and 400.

printOnQuit (class **PrintOnQuit**)

If *True* the currently selected font name is printed to standard output when the quit button is activated. Equivalent to the **-print** option.

sampleText (class **Text**)

The sample 1-byte text to use for linearly indexed fonts. Each glyph index is a single byte, with newline separating lines.

sampleText16 (class **Text16**)

The sample 2-byte text to use for matrix-encoded fonts. Each glyph index is two bytes, with a 1-byte newline separating lines.

scaledFonts (class **ScaledFonts**)

If *True* then selection of arbitrary pixel and point sizes for scalable fonts is enabled.

Widget specific resources:

showUnselectable (class **ShowUnselectable**)

Specifies, for each field menu, whether or not to show values that are not currently selectable, based upon previous field selections. If shown, the unselectable values are clearly identified as such and do not highlight when the pointer is moved down the menu. The full name of this resource is

fieldN.menu.options.showUnselectable, class
MenuButton.SimpleMenu.Options.ShowUnselectable; where N is replaced with the field number (starting with the left-most field numbered 0). The default is True for all but field 11 (average width of characters in font) and False for field 11. If you never want to see unselectable entries, '*menu.options.showUnselectable:False' is a reasonable thing to specify in a resource file.

| | |
|------------------|--|
| FILES | \$XFILESEARCHPATH/XFontSel |
| SEE ALSO | xrdb(1), xfd(1) |
| BUGS | <p>Sufficiently ambiguous patterns can be misinterpreted and lead to an initial selection string which may not correspond to what the user intended and which may cause the initial sample text output to fail to match the proffered string. Selecting any new field value will correct the sample output, though possibly resulting in no matching font.</p> <p>Should be able to return a FONT for the PRIMARY selection, not just a STRING.</p> <p>Any change in a field value will cause xfontsel to assert ownership of the PRIMARY_FONT selection. Perhaps this should be parameterized.</p> <p>When running on a slow machine, it is possible for the user to request a field menu before the font names have been completely parsed. An error message indicating a missing menu is printed to stderr but otherwise nothing bad (or good) happens.</p> <p>The average-width menu is too large to be useful.</p> |
| COPYRIGHT | Copyright 1989, 1991 by the Massachusetts Institute of Technology See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Ralph R. Swick, Digital Equipment Corporation/MIT Project Athena |

| | |
|--------------------|--|
| NAME | xgc – X graphics demo |
| SYNOPSIS | xgc [-toolkitoption] |
| DESCRIPTION | <p>The xgc program demonstrates various features of the X graphics primitives. In X, most of the details about the graphics to be generated are stored in a resource called a graphics context (GC). The xgc program provides a user interface for setting various GC components. Pressing the "Run" button causes these results to be displayed in the large drawing window on the right. Timing information is displayed in the window immediately below.</p> <p>The items in the upper left hand window work as follows:</p> <p><i>Function</i> – specify the logical function with which primitives will drawn. The most usual setting is "set", i.e. simply to render pixels without regard to what has been drawn before.</p> <p><i>LineStyle</i> – specify whether lines should be drawn solid in foreground, dashed in foreground or alternating foreground and background.</p> <p><i>CapStyle</i> – specify the appearance of the ends of a line.</p> <p><i>JoinStyle</i> – specify the appearance of joints between consecutive lines drawn within a single graphics primitive.</p> <p>solid, tiled with a pixmap or stippled.</p> <p>rule means that if areas overlap an odd number of times, they are not drawn. Winding rule means that overlapping areas are always filled, regardless of how many times they overlap.</p> <p><i>ArcMode</i> – specifies the rule for filling of arcs. The boundary of the arc is either a Chord or two radii.</p> <p><i>planemask</i> – specifies which planes of the drawing window are modified. By default, all planes are modified.</p> <p><i>dashlist</i> – specifies a pattern to be used when drawing dashed lines.</p> <p><i>Line Width</i> – specifies the width in pixels of lines to be drawn. Zero means to draw using the server's fastest algorithm with a line width of one pixel.</p> <p><i>Font</i> – specifies the font to be used for text primitives.</p> <p><i>Foreground</i> and <i>Background</i> – specify the pixel values to be applied when drawing primitives. The Foreground value is used as the pixel value for set bits in the source in all primitives. The Background value is used as the pixel value for unset bits in the source when using Copy Plane, drawing lines with LineStyle of DoubleDash and filling with FillStyle of OpaqueStippled.</p> <p><i>Percentage of Test</i> – scrollbar permits specifying only a percentage of the test to be run. The number at the left indicates the current setting, which defaults to 100%.</p> <p>The window labeled "Test" permits choice of one a number of graphics primitive tests, including Points, Segments, Lines, Arcs and Filled Arcs, 8-bit Text and Image Text, Rectangles and Filled Rectangles, Image draws, as well as Copy Plane and Copy Area.</p> |

The window to the right of this has buttons which permit record/playback of the primitives rendered.

OPTIONS

Xgc accepts all of the standard X Toolkit command line options.

X DEFAULTS

This program accepts the usual defaults for toolkit applications.

ENVIRONMENT**DISPLAY**

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the `RESOURCE_MANAGER` property.

SEE ALSO

X11(7),
O'Reilly Xlib Programming Manual, Volume I, Chapter 5

BUGS

This program isn't really finished yet. In particular, buttons whose labels appear in parentheses indicate features which are not yet implemented.

COPYRIGHT

Copyright 1989, Massachusetts Institute of Technology.

AUTHORS

Dan Schmidt, MIT

| | |
|--------------------|--|
| NAME | xhost – server access control program for X |
| SYNOPSIS | xhost [[+] <i>-</i>] <i>name</i> ...] |
| DESCRIPTION | <p>The xhost program is used to add and delete host names or user names to the list allowed to make connections to the X server. In the case of hosts, this provides a rudimentary form of privacy control and security. It is only sufficient for a workstation (single user) environment, although it does limit the worst abuses. Environments which require more sophisticated measures should implement the user-based mechanism, or use the hooks in the protocol for passing other authentication data to the server.</p> <p>Hostnames that are followed by two colons (::) are used in checking DECnet connections; all other hostnames are used for TCP/IP connections.</p> <p>User names contain an at-sign (@). When Secure RPC is being used, the network independent netname (e.g., "unix.uid@domainname") can be specified, or a local user can be specified with just the username and a trailing at-sign (e.g., "joe@").</p> |
| OPTIONS | <p>Xhost accepts the following command line options described below. For security, the options that effect access control may only be run from the "controlling host". For workstations, this is the same machine as the server. For X terminals, it is the login host.</p> <p>[+]<i>name</i> The given <i>name</i> (the plus sign is optional) is added to the list allowed to connect to the X server. The name can be a host name or a user name.</p> <p><i>-name</i> The given <i>name</i> is removed from the list of allowed to connect to the server. The name can be a host name or a user name. Existing connections are not broken, but new connection attempts will be denied. Note that the current machine is allowed to be removed; however, further connections (including attempts to add it back) will not be permitted. Resetting the server (thereby breaking all connections) is the only way to allow local connections again.</p> <p>+ Access is granted to everyone, even if they aren't on the list (i.e., access control is turned off).</p> <p>- Access is restricted to only those on the list (i.e., access control is turned on).</p> <p><i>nothing</i> If no command line arguments are given, a message indicating whether or not access control is currently enabled is printed, followed by the list of those allowed to connect. This is the only option that may be used from machines other than the controlling host.</p> |
| DIAGNOSTICS | For each name added to the access control list, a line of the form " <i>name</i> being added to access contro list" is printed. For each name removed from the access control list, a line of the form " <i>name</i> being removed from access contro list" is printed. |
| FILES | /etc/X*.hosts |

SEE ALSO X11(7), Xserver(1), xdm(1)

ENVIRONMENT **DISPLAY**
to get the default host and display to use.

BUGS You can't specify a display on the command line because **-display** is a valid command line argument (indicating that you want to remove the machine named "*display*" from the access list).

This is not really a bug, but the X server stores network addresses, not host names. If somehow you change a host's network address while the server is still running, *xhost* must be used to add the new address and/or remove the old address.

COPYRIGHT Copyright 1988, Massachusetts Institute of Technology.
See X11(7) for a full statement of rights and permissions.

AUTHORS Bob Scheifler, MIT Laboratory for Computer Science,
Jim Gettys, MIT Project Athena (DEC).

| | |
|--------------------|---|
| NAME | xinit – X Window System initializer |
| SYNOPSIS | xinit [[<i>client</i>] <i>options</i>] [-- [<i>server</i>] [<i>display</i>] <i>options</i>] |
| DESCRIPTION | <p>The xinit program is used to start the X Window System server and a first client program on systems that cannot start X directly from <i>/etc/init</i> or in environments that use multiple window systems. When this first client exits, xinit will kill the X server and then terminate.</p> <p>If no specific client program is given on the command line, xinit will look for a file in the user's home directory called <i>.xinitrc</i> to run as a shell script to start up client programs. If no such file exists, xinit will use the following as a default:</p> <pre style="margin-left: 40px;">xterm -geometry +1+1 -n login -display :0</pre> <p>If no specific server program is given on the command line, xinit will look for a file in the user's home directory called <i>.xserverrc</i> to run as a shell script to start up the server. If no such file exists, xinit will use the following as a default:</p> <pre style="margin-left: 40px;">X :0</pre> <p>Note that this assumes that there is a program named <i>X</i> in the current search path. However, servers are usually named <i>Xdisplaytype</i> where <i>displaytype</i> is the type of graphics display which is driven by this server. The site administrator should, therefore, make a link to the appropriate type of server on the machine, or create a shell script that runs xinit with the appropriate server.</p> <p>An important point is that programs which are run by <i>.xinitrc</i> should be run in the background if they do not exit right away, so that they don't prevent other programs from starting up. However, the last long-lived program started (usually a window manager or terminal emulator) should be left in the foreground so that the script won't exit (which indicates that the user is done and that xinit should exit).</p> <p>An alternate client and/or server may be specified on the command line. The desired client program and its arguments should be given as the first command line arguments to xinit. To specify a particular server command line, append a double dash (--) to the xinit command line (after any client and arguments) followed by the desired server command.</p> <p>Both the client program name and the server program name must begin with a slash (/) or a period (.). Otherwise, they are treated as an arguments to be appended to their respective startup lines. This makes it possible to add arguments (for example, foreground and background colors) without having to retype the whole command line.</p> <p>If an explicit server name is not given and the first argument following the double dash (--) is a colon followed by a digit, xinit will use that number as the display number instead of zero. All remaining arguments are appended to the server command line.</p> |

EXAMPLES

Below are several examples of how command line arguments in **xinit** are used.

xinit This will start up a server named *X* and run the user's *.xinitrc*, if it exists, or else start an *xterm*.

xinit -- /usr/bin/X11/Xqdss :1

This is how one could start a specific type of server on an alternate display.

xinit -geometry =80x65+10+10 -fn 8x13 -j -fg white -bg navy

This will start up a server named *X*, and will append the given arguments to the default *xterm* command. It will ignore *.xinitrc*.

xinit -e widgets -- ./Xsun -l -c

This will use the command *./Xsun -l -c* to start the server and will append the arguments *-e widgets* to the default *xterm* command.

xinit /usr/ucb/rsh fasthost cpupig -display ws:1 -- :1 -a 2 -t 5

This will start a server named *X* on display 1 with the arguments *-a 2 -t 5*. It will then start a remote shell on the machine **fasthost** in which it will run the command *cpupig*, telling it to display back on the local workstation.

Below is a sample *.xinitrc* that starts a clock, several terminals, and leaves the window manager running as the "last" application. Assuming that the window manager has been configured properly, the user then chooses the "Exit" menu item to shut down *X*.

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

Sites that want to create a common startup environment could simply create a default *.xinitrc* that references a site-wide startup file:

```
#!/bin/sh
. /usr/local/lib/site.xinitrc
```

Another approach is to write a script that starts **xinit** with a specific shell script. Such scripts are usually named *x11*, *xstart*, or *startx* and are a convenient way to provide a simple interface for novice users:

```
#!/bin/sh
xinit /usr/local/lib/site.xinitrc -- /usr/bin/X11/X bc
```

**ENVIRONMENT
VARIABLES****DISPLAY** This variable gets set to the name of the display to which clients should connect.**XINITRC** This variable specifies an init file containing shell commands to start up the initial windows. By default, *.xinitrc* in the home directory will be used.**FILES***.xinitrc* default client script*xterm* client to run if *.xinitrc* does not exist*.xserverrc* default server script*X* server to run if *.xserverrc* does not exist**SEE ALSO****olwm(1), openwin(1), props(1), X11(7), Xserver(1), xterm(1)****COPYRIGHT**Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.**AUTHOR**

Bob Scheifler, MIT Laboratory for Computer Science

| | |
|--------------------|---|
| NAME | xkill – kill a client by its X resource |
| SYNOPSIS | xkill [-display <i>displayname</i>] [-id <i>resource</i>] [-button <i>number</i>] [-frame] [-all] |
| DESCRIPTION | Xkill is a utility for forcing the X server to close connections to clients. This program is very dangerous, but is useful for aborting programs that have displayed undesired windows on a user's screen. If no resource identifier is given with <i>-id</i> , xkill will display a special cursor as a prompt for the user to select a window to be killed. If a pointer button is pressed over a non-root window, the server will close its connection to the client that created the window. |
| OPTIONS | <p>-display <i>displayname</i> This option specifies the name of the X server to contact.</p> <p>-id <i>resource</i> This option specifies the X identifier for the resource whose creator is to be aborted. If no resource is specified, xkill will display a special cursor with which you should select a window to be kill.</p> <p>-button <i>number</i> This option specifies the number of pointer button that should be used in selecting a window to kill. If the word "any" is specified, any button on the pointer may be used. By default, the first button in the pointer map (which is usually the leftmost button) is used.</p> <p>-all This option indicates that all clients with top-level windows on the screen should be killed. Xkill will ask you to select the root window with each of the currently defined buttons to give you several chances to abort. Use of this option is highly discouraged.</p> <p>-frame This option indicates that xkill should ignore the standard conventions for finding top-level client windows (which are typically nested inside a window manager window), and simply believe that you want to kill direct children of the root.</p> |
| XDEFAULTS | Button Specifies a specific pointer button number or the word "any" to use when selecting windows. |
| SEE ALSO | X11(7) , xwininfo(1) , XKillClient and XGetPointerMapping in the Xlib Programmers Manual, KillClient in the X Protocol Specification |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium Dana Chee, Bellcore |

| | |
|--------------------|---|
| NAME | xload – system load average display for X |
| SYNOPSIS | xload [<i>toolkitoptions</i>] [-scale <i>integer</i>] [-update <i>seconds</i>] [-hl <i>color</i>] [-highlight <i>color</i>] [-jumpscroll <i>pixels</i>] [-label <i>string</i>] [-nolabel] [-lights] |
| DESCRIPTION | The xload program displays a periodically updating histogram of the system load average. |
| OPTIONS | <p>Xload accepts all of the standard X Toolkit command line options (see Xsun(1)). The order of the options is unimportant. Xload also accepts the following additional options:</p> <p>-hl <i>color</i> or -highlight <i>color</i> This option specifies the color of the scale lines.</p> <p>-jumpscroll <i>pixels</i> The number of pixels to shift the graph to the left when the graph reaches the right edge of the window. The default value is 1/2 the width of the current window. Smooth scrolling can be achieved by setting it to 1.</p> <p>-label <i>string</i> The string to put into the label above the load average.</p> <p>-nolabel If this command line option is specified then no label will be displayed above the load graph.</p> <p>-lights When specified, this option causes xload to display the current load average by using the keyboard leds; for a load average of <i>n</i>, xload lights the first <i>n</i> keyboard leds. This option turns off the usual screen display.</p> <p>-scale <i>integer</i> This option specifies the minimum number of tick marks in the histogram, where one division represents one load average point. If the load goes above this number, xload will create more divisions, but it will never use fewer than this number. The default is 1.</p> <p>-update <i>seconds</i> This option specifies the interval in seconds at which xload updates its display. The minimum amount of time allowed between updates is 1 second. The default is 10.</p> |
| RESOURCES | <p>In addition to the resources available to each of the widgets used by xload there is one resource defined by the application itself.</p> <p>showLabel (class Boolean) If False then no label will be displayed.</p> |
| WIDGETS | In order to specify resources, it is useful to know the hierarchy of the widgets which compose xload . In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name. |

XLoad xload
 Paned paned
 Label label
 StripChart load

ENVIRONMENT**DISPLAY**

to get the default host and display number.

XENVIRONMENT

to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

FILES

/usr/openwin/lib/app-defaults/XLoad - specifies required resources

SEE ALSO

X11(7), **xrdb(1)**, **mem(7)**, Athena StripChart Widget.

BUGS

This program requires the ability to open and read the special system file */dev/kmem*. Sites that do not allow general access to this file should make **xload** belong to the same group as */dev/kmem* and turn on the *set group id* permission flag.

Reading */dev/kmem* is inherently non-portable. Therefore, the routine used to read it (*get_load.c*) must be ported to each new operating system.

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
 See **X11(7)** for a full statement of rights and permissions.

AUTHORS

K. Shane Hartman (MIT-LCS) and Stuart A. Malone (MIT-LCS);
 with features added by Jim Gettys (MIT-Athena), Bob Scheifler (MIT-LCS), Tony Della Fera (MIT-Athena), and Chris Peterson (MIT-LCS).

| | |
|--------------------|---|
| NAME | xlock – Locks the local X display until a password is entered |
| SYNOPSIS | xlock [-display <i>dsp</i>] [-help] [-name <i>resource-name</i>] [-resources] [-/+remote] [-/+mono] [-/+nolock] [-/+allowroot] [-/+enablesaver] [-/+allowaccess] [-/+echokeys] [-/+usefirst] [-/+v] [-delay <i>usecs</i>] [-batchcount <i>num</i>] [-nice <i>level</i>] [-timeout <i>seconds</i>] [-saturation <i>value</i>] [-font <i>fontname</i>] [-bg <i>color</i>] [-fg <i>color</i>] [-mode <i>modename</i>] [-username <i>textstring</i>] [-password <i>textstring</i>] [-info <i>textstring</i>] [-validate <i>textstring</i>] [-invalid <i>textstring</i>] |
| DESCRIPTION | <p>xlock locks the X server till the user enters their password at the keyboard. While xlock is running, all new server connections are refused. The screen saver is disabled. The mouse cursor is turned off. The screen is blanked and a changing pattern is put on the screen. If a key or a mouse button is pressed then the user is prompted for the password of the user who started xlock.</p> <p>If the correct password is typed, then the screen is unlocked and the X server is restored. When typing the password Control-U and Control-H are active as kill and erase respectively. To return to the locked screen, click in the small icon version of the changing pattern.</p> |
| OPTIONS | <p>-display <i>dsp</i> The <i>display</i> option sets the X11 display to lock. xlock locks all available screens on a given server, and restricts you to locking only a local server such as unix:0, localhost:0, or :0 unless you set the -remote option.</p> <p>-name <i>resource-name</i> <i>resource-name</i> is used instead of XLock when looking for resources to configure xlock.</p> <p>-mode <i>modename</i> As of this writing there are eight display modes supported (plus one more for random selection of one of the eight).</p> <p>hop Hop mode shows the "real plane fractals" from the September 1986 issue of Scientific American.</p> <p>life Life mode shows Conway's game of life.</p> <p>qix Qix mode shows the spinning lines similar to the old video game by the same name.</p> <p>image Image mode shows several sun logos randomly appearing on the screen.</p> <p>swarm Swarm mode shows a swarm of bees following a wasp.</p> <p>rotor Rotor mode shows a swirling rotorlike thing.</p> <p>pyro Pyro mode shows fireworks.</p> |

flame Flame mode shows wierd but cool fractals.

blank Blank mode shows nothing but a black screen.

random Random mode picks a random mode from all of the above except blank mode.

-delay *usecs*

The *delay* option sets the speed at which a mode will operate. It simply sets the number of microseconds to delay between batches of animations. In blank mode, it is important to set this to some small number of seconds, because the keyboard and mouse are only checked after each delay, so you cannot set the delay too high, but a delay of zero would needlessly consume cpu checking for mouse and keyboard input in a tight loop, since blank mode has no work to do.

-batchcount *num*

The *batchcount* option sets number of *things* to do per batch to *num* . In hop mode this refers to the number of pixels rendered in the same color. In life mode it is the number of generations to let each species live. In qix mode it is the number of lines rendered in the same color. In image mode it is the number of sunlogos on screen at once. In swarm mode it is the number of bees. In rotor mode it is the number of rotor thingys which whirr... In pyro mode it is the maximum number flying rockets at one time. In flame mode it is the number of levels to recurse (larger = more complex). In blank mode it means nothing.

-nice *nicelevel*

The *nice* option sets system nicelevel of the **xlock** process to *nicelevel* .

-timeout *seconds*

The *timeout* option sets the number of *seconds* before the password screen will time out.

-saturation *value*

The *saturation* option sets saturation of the color ramp used to *value* . 0 is grayscale and 1 is very rich color. 0.4 is a nice pastel.

-font *fontname*

The *font* option sets the font to be used on the prompt screen.

-fg *color*

The *fg* option sets the color of the text on the password screen to *color* .

-bg *color*

The *bg* option sets the color of the background on the password screen to *color* .

-username *textstring*

textstring is shown in front of user name, defaults to "Name: ".

-password *textstring*

textstring is the password prompt string, defaults to "Password: ".

-info *textstring*

textstring is an informational message to tell the user what to do, defaults to "Enter

password to unlock; select icon to lock."

-validate *textstring*
textstring **-validate** *message shown while validating the password, defaults to "Validating login..."*

-invalid *textstring*
textstring **-invalid** *message shown when password is invalid, defaults to "Invalid login."*

-resources
The *resources* option prints the default resource file for **xlock** to standard output.

-/+remote
The *remote* option tells **xlock** to not stop you from locking remote X11 servers. This option should be used with care and is intended mainly to lock X11 terminals which cannot run **xlock** locally. If you lock someone else's workstation, they will have to know **your** password to unlock it. Using *+remote* overrides any resource derived values for *remote* and prevents **xlock** from being used to lock other X11 servers. (Use '+' instead of '-' to override resources for other options that can take the '+' modifier similarly.)

-/+mono
The *mono* option causes **xlock** to display monochrome, (black and white) pixels rather than the default colored ones on color displays.

+/-nolock
The *nolock* option causes **xlock** to only draw the patterns and not lock the display. A keypress or a mouse click will terminate the screen saver.

-/+allowroot
The *allowroot* option allows the root password to unlock the server as well as the user who started **xlock**.

-/+enablesaver
By default **xlock** will disable the normal X server's screen saver since it is in effect a replacement for it. Since it is possible to set delay parameters long enough to cause phosphor burn on some displays, this option will turn back on the default screen-saver which is very careful to keep most of the screen black.

-/+allowaccess
This option is required for servers which do not allow clients to modify the host access control list. It is also useful if you need to run x clients on a server which is locked for some reason... When *allowaccess* is true, the X11 server is left open for clients to attach and thus lowers the inherent security of this lockscreen. A side effect of using this option is that if **xlock** is killed -KILL, the access control list is not lost.

-/+echokeys
The *echokeys* option causes **xlock** to echo '?' characters for each key typed into the password prompt. Some consider this a security risk, so the default is to not echo anything.

-/+usefirst

The *usefirst* option causes **xlock** to use the keystroke which got you to the password screen as the first character in the password. The default is to ignore the first key pressed.

-v Verbose mode, tells what options it is going to use.

****WARNING****

xlock can appear to hang if it is competing with a high-priority process for the CPU. For example, if **xlock** is started after a process with 'nice -20' (high priority), **xlock** will take considerable amount of time to respond.

BUGS

"kill -KILL **xlock** " causes the server that was locked to be unusable, since all hosts (including localhost) were removed from the access control list to lock out new X clients, and since **xlock** couldn't catch SIGKILL, it terminated before restoring the access control list. This will leave the X server in a state where *"you can no longer connect to that server, and this operation cannot be reversed unless you reset the server."*

-From the X11R4 Xlib Documentation, Chapter 7.

SEE ALSO

Xlib Documentation.

AUTHOR

Patrick J. Naughton

COPYRIGHT

Copyright (c) 1988-91 by Patrick J. Naughton and Sun Microsystems, Inc.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

| | |
|--------------------|---|
| NAME | xlogo – X Window System logo |
| SYNOPSIS | xlogo [- <i>toolkitoption</i> ...] |
| DESCRIPTION | The xlogo program displays the X Window System logo. This program is nothing more than a wrapper around the <i>undocumented</i> Athena Logo widget. |
| OPTIONS | <p>Xlogo accepts all of the standard X Toolkit command line options, as well as the following:</p> <p>-shape This option indicates that the logo window should be shaped rather than rectangular.</p> |
| RESOURCES | <p>The default width and the default height are each 100 pixels. This program uses the <i>Logo</i> widget in the Athena widget set. It understands all of the Simple widget resource names and classes as well as:</p> <p>foreground (class Foreground) Specifies the color for the logo. The default is depends on whether <i>reverseVideo</i> is specified. If <i>reverseVideo</i> is specified the default is <i>XtDefaultForeground</i>, otherwise the default is <i>XtDefaultBackground</i>.</p> <p>shapeWindow (class ShapeWindow) Specifies that the window is shaped to the X logo. The default is False.</p> |
| WIDGETS | <p>In order to specify resources, it is useful to know the hierarchy of the widgets which compose xlogo. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.</p> <pre>XLogo xlogo Logo xlogo</pre> |
| ENVIRONMENT | <p>DISPLAY to get the default host and display number.</p> <p>XENVIRONMENT to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.</p> |
| FILES | /usr/openwin/lib/app-defaults/XLogo - specifies required resources |
| SEE ALSO | X11(7) , xrdb(1) |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |

AUTHORS

Ollie Jones of Apollo Computer and Jim Fulton of the MIT X Consortium wrote the logo graphics routine, based on a graphic design by Danny Chong and Ross Chapman of Apollo Computer.

| | |
|--------------------|---|
| NAME | xlsatoms – list interned atoms defined on server |
| SYNOPSIS | xlsatoms [-options ...] |
| DESCRIPTION | 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. |
| OPTIONS | <p>–display <i>dpy</i> This option specifies the X server to which to connect.</p> <p>–format <i>string</i> This option specifies a <i>printf</i>-style string used to list each atom <i><value,name></i> pair, printed in that order (<i>value</i> is an <i>unsigned long</i> and <i>name</i> is a <i>char *</i>). Xlsatoms will supply a newline at the end of each line. The default is <i>%ld\t%s</i>.</p> <p>–range [<i>low</i>]-[<i>high</i>] This option specifies the range of atom values to check. If <i>low</i> is not given, a value of 1 assumed. If <i>high</i> is not given, <i>xlsatoms</i> will stop at the first undefined atom at or above <i>low</i>.</p> <p>–name <i>string</i> This option specifies the name of an atom to list. If the atom does not exist, a message will be printed on the standard error.</p> |
| SEE ALSO | X11(7) , Xserver(1) , xprop(1) |
| ENVIRONMENT | DISPLAY to get the default host and display to use. |
| COPYRIGHT | Copyright 1989, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium |

| | |
|--------------------|--|
| NAME | <code>xlsclients</code> – list client applications running on a display |
| SYNOPSIS | <code>xlsclients</code> [-display <i>displayname</i>] [-a] [-l] [-m <i>maxcmdlen</i>] |
| DESCRIPTION | <code>Xlsclients</code> is a utility for listing information about the client applications running on a display. It may be used to generate scripts representing a snapshot of the the user's current session. |
| OPTIONS | <p>-display <i>displayname</i> This option specifies the X server to contact.</p> <p>-a This option indicates that clients on all screens should be listed. By default, only those clients on the default screen are listed.</p> <p>-l This option indicates that a long listing showing the window name, icon name, and class hints in addition to the machine name and command string shown in the default listing.</p> <p>-m <i>maxcmdlen</i> This option specifies the maximum number of characters in a command to print out. The default is 10000.</p> |
| ENVIRONMENT | <p>DISPLAY To get the default host, display number, and screen.</p> |
| SEE ALSO | <code>X11(7)</code> , <code>xwininfo(1)</code> , <code>xprop(1)</code> |
| COPYRIGHT | Copyright 1989, Massachusetts Institute of Technology. See <code>X11(7)</code> for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium |

| | |
|--------------------|---|
| NAME | xlsfonts – server font list displayer for X |
| SYNOPSIS | xlsfonts [-options ...] [-fn pattern] |
| DESCRIPTION | <p>Xlsfonts lists the fonts that match the given <i>pattern</i>. The wildcard character "*" may be used to match any sequence of characters (including none), and "?" to match any single character. If no pattern is given, "*" is assumed.</p> <p>The "*" and "?" characters must be quoted to prevent them from being expanded by the shell.</p> |
| OPTIONS | <p>-display <i>host:dpy</i> This option specifies the X server to contact.</p> <p>-l[l[l]] This option indicates that medium, long, and very long listings, respectively, should be generated for each font.</p> <p>-m This option indicates that long listings should also print the minimum and maximum bounds of each font.</p> <p>-C This option indicates that listings should use multiple columns. This is the same as -n 0.</p> <p>-1 This option indicates that listings should use a single column. This is the same as -n 1.</p> <p>-w <i>width</i> This option specifies the width in characters that should be used in figuring out how many columns to print. The default is 79.</p> <p>-n <i>columns</i> This option specifies the number of columns to use in displaying the output. By default, it will attempt to fit as many columns of font names into the number of character specified by -w <i>width</i>.</p> <p>-u This option indicates that the output should be left unsorted.</p> <p>-o This option indicates that xlsfonts should do an OpenFont (and QueryFont, if appropriate) rather than a ListFonts. This is useful if ListFonts or ListFontsWithInfo fail to list a known font (as is the case with some scaled font systems).</p> |
| SEE ALSO | X11(7) , Xserver(1) , xset(1) , xfd(1) |
| ENVIRONMENT | <p>DISPLAY to get the default host and display to use.</p> |
| BUGS | Doing "xlsfonts -l" can tie up your server for a very long time. This is really a bug with single-threaded non-preemptable servers, not with this program. |

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Mark Lillibridge, MIT Project Athena; Jim Fulton, MIT X Consortium; Phil Karlton, SGI

| | |
|--------------------|--|
| NAME | <code>xlswins</code> – server window list displayer for X |
| SYNOPSIS | <code>xlswins</code> [-options ...] [<i>windowid</i> ...] |
| DESCRIPTION | <code>xlswins</code> lists the window tree. By default, the root window is used as the starting point, although a specific window may be specified using the <i>-id</i> option. If no specific windows are given on the command line, the root window will be used. |
| OPTIONS | <p>-display <i>displayname</i> This option specifies the X server to contact.</p> <p>-l This option indicates that a long listing should be generated for each window. This includes a number indicating the depth, the geometry relative to the parent as well as the location relative to the root window.</p> <p>-format <i>radix</i> This option specifies the radix to use when printing out window ids. Allowable values are: <i>hex</i>, <i>octal</i>, and <i>decimal</i>. The default is <i>hex</i>.</p> <p>-indent <i>number</i> This option specifies the number of spaces that should be indented for each level in the window tree. The default is 2.</p> |
| SEE ALSO | <code>xprop</code> (1), <code>xwininfo</code> (1) |
| ENVIRONMENT | <p>DISPLAY to get the default host and display to use.</p> |
| BUGS | This should be integrated with <code>xwininfo</code> somehow. |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See <i>X11(7)</i> for a full statement of rights and permissions. |
| AUTHOR | Jim Fulton, MIT X Consortium |

| | |
|--------------------|---|
| NAME | xmac – display Apple MacPaint image files under X windows |
| SYNOPSIS | xmac <i>filename</i> [-ps] [<i>host:display</i>] [-geometry] |
| DESCRIPTION | xmac displays a MacPaint file in a window, allows resize/move, and has an icon. xmac will send the Postscript commands to print the image to standard out if you include the command line option, -ps . xmac accepts two other optional command line arguments. You may specify a display name in the form <i>host:display</i> (see X11(7)). And you may provide a geometry specification. If you don't give a geometry specification, xmac will ask you where you want to put the window when it starts up. See X11(7) for a full explanation. |
| BUGS | There are no known bugs. There are lots of lacking features. I would like to add editing capability in the future, along with the ability to clip part of an image to a bitmap format file; as well as replacing the desktop pattern with an image. Also there should be a way to kill the process, i.e. a keypress or a mouse click in a box. Also a title bar would be nice. |
| ENVIRONMENT | XMAC - the default directory for searching for image files, (after "."). |
| SEE ALSO | X11(7) , Xlib Documentation. |
| AUTHOR | Copyright (c) 1987 by Patrick J. Naughton, (naughton@sun.soe.clarkson.edu) Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. |

| | |
|--------------------|---|
| NAME | xmag – magnify parts of the screen |
| SYNOPSIS | xmag [-mag <i>magfactor</i>] [-source <i>geom</i>] [-toolkitoption . . .] |
| DESCRIPTION | <p>The xmag program allows you to magnify portions of an X screen. If no explicit region is specified, a square with the pointer in the upper left corner is displayed indicating the area to be enlarged. The area can be dragged out to the desired size by pressing Button 2. Once a region has been selected, a window is popped up showing a blown up version of the region in which each pixel in the source image is represented by a small square of the same color. Pressing Button1 in the enlargement window shows the position and RGB value of the pixel under the pointer until the button is released. Typing “Q” or “^C” in the enlargement window exits the program. The application has 5 buttons across its top. <i>Close</i> deletes this particular magnification instance. <i>Replace</i> brings up the rubber band selector again to select another region for this magnification instance. <i>New</i> brings up the rubber band selector to create a new magnification instance. <i>Cut</i> puts the magnification image into the primary selection. <i>Paste</i> copies the primary selection buffer into xmag. Note that you can cut and paste between xmag and the bitmap(1) program. Resizing xmag resizes the magnification area. xmag preserves the colormap, visual, and window depth of the source.</p> |
| WIDGETS | <p>xmag uses the X Toolkit and the Athena Widget Set. The magnified image is displayed in the Scale widget. For more information, see the Athena Widget Set documentation. Below is the widget structure of the xmag application. Indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.</p> <pre style="margin-left: 40px;"> Xmag xmag RootWindow root TopLevelShell xmag Paned pane1 Paned pane2 Command close Command replace Command new Command select Command paste Label xmag label Paned pane2 Scale scale OverrideShell pixShell Label pixLabel </pre> |
| OPTIONS | <p>-source <i>geom</i> This option specifies the size and/or location of the source region on the screen. By default, a 64x64 square is provided for the user to select an area of the screen.</p> |

-mag *integer* This option indicates the magnification to be used. 5 is the default.

COPYRIGHT

Copyright 1991, Massachusetts Institute of Technology.
See **X11(7)**, for a full statement of rights and permissions.

SEE ALSO

X11(7), **xmag_multivis(6)**

AUTHORS

Dave Sternlicht and Davor Matic, MIT X Consortium.

| | |
|--------------------|---|
| NAME | xmag_multivis – magnify parts of the screen |
| SYNOPSIS | xmag_multivis [-option ...] |
| DESCRIPTION | <p>The xmag_multivis program allows you to magnify portions of the screen. If no explicit region is specified, a square centered around the pointer is displayed indicating the area to be enlarged. Once a region has been selected, a window is popped up showing a blown up version of the region in which each pixel in the source image is represented by a small square of the same color. Pressing Button1 on the pointer in the enlargement window pops up a small window displaying, in hexadecimal notation, the position, pixel value, and RGB value of the pixel under the pointer until the button is released. Pressing the space bar or any other pointer button removes the enlarged image so that another region may be selected. Pressing “q”, “Q”, or “^C” in the enlargement window exits the program.</p> <p>On displays that export multiple Visuals, the selected region is first examined for overlaps by windows of differing depth, visual or colormap. If the selected region does contain such windows, the image displayed in the enlargement window is constructed by compositing the images from the various windows under the selected region. In this case, the composite image will only be displayed in a TrueColor 24 bit deep window, provided such a window can be created on that screen. The image displayed in the enlargement window will exhibit true colors for every pixel in the composite image, regardless of the Colormap state (Installed/UnInstalled) of any of the windows within the selected region.</p> |
| OPTIONS | <p>-display <i>display</i> This option specifies the X server to use for both reading the screen and displaying the enlarged version of the image.</p> <p>-geometry <i>geom</i> This option specifies the size and/or location of the enlargement window. By default, the size is computed from the size of the source region and the desired magnification. Therefore, only one of -source <i>size</i> and -mag <i>magfactor</i> options may be specified if a window size is given with this option.</p> <p>-source <i>geom</i> This option specifies the size and/or location of the source region on the screen. By default, a 64x64 square centered about the pointer is provided for the user to select an area of the screen. The size of the source is used with the desired magnification to compute the default enlargement window size. Therefore, only one of -geometry <i>size</i> and -mag <i>magfactor</i> options may be specified if a source size is given with this option.</p> <p>-mag <i>magfactor</i> This option specifies an integral factor by which the source region should be enlarged. The default magnification is 5. This is used with the size of the source to compute the default enlargement window size. Therefore, only one of -geometry <i>size</i> and -source <i>geom</i> options may be specified if a magnification</p> |

factor is given with this option.

-bw *pixels*

This option specifies the width in pixels of the border surrounding the enlargement window.

-bd *color*

This option specifies the color to use for the border surrounding the enlargement window.

-bg *colororpixelvalue*

This option specifies the name of the color to be used as the background of the enlargement window. If the name begins with a percent size (%), it is interpreted to be an absolute pixel value. This is useful when displaying large areas since pixels that are the same color as the background do not need to be painted in the enlargement. The default is to use the BlackPixel of the screen.

-fn *fontname*

This option specifies the name of a font to use when displaying pixel values (used when Button1 is pressed in the enlargement window).

-z

This option indicates that the server should be grabbed during the dynamics and the call to XGetImage. This is useful for ensuring that clients don't change their state as a result of entering or leaving them with the pointer.

X DEFAULTS

The **xmag_multivis** program uses the following X resources:

geometry (class **Geometry**)

Specifies the size and/or location of the enlargement window.

source (class **Source**)

Specifies the size and/or location of the source region on the screen.

magnification (class **Magnification**)

Specifies the enlargement factor.

borderWidth (class **BorderWidth**)

Specifies the border width in pixels.

borderColor (class **BorderColor**)

Specifies the color of the border.

background (class **Background**)

Specifies the color or pixel value to be used for the background of the enlargement window.

font (class **Font**)

Specifies the name of the font to use when displaying pixel values when the user presses Button1 in the enlargement window.

SEE ALSO

xwd(1), **xmag(1)**

BUGS

On displays that export multiple visuals, if the selected region needs to be composited from each window, **xmag_multivis** insists on a TrueColor 24 bit window to display the selected region, and exits with a failure if such a window cannot be created. It would be nice, instead if **xmag_multivis** would determine the best possible visual to display the image, or instead tried to display the image in a visual selected by the user in a command line option.

Because the window size equals the source size times the magnification, you only need to specify two of the three parameters. This can be confusing.

Being able to drag the pointer around and see a dynamic display would be very nice.

Another possible interface would be for the user to drag out the desired area to be enlarged.

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.

MultiVisual code: Copyright (c) 1990-92 by Sun Microsystems, Inc.

AUTHOR

Jim Fulton, MIT X Consortium

MultiVisual code: Milind Pansare, Sun Microsystems, Inc.

| | |
|--------------------|--|
| NAME | xmakemap – make a keyboard mapping to be used as input to xmodmap |
| SYNOPSIS | xmakemap |
| DESCRIPTION | <p>xmakemap will produce a keyboard mapping in a form that is suitable as input to the xmodmap(1) command. xmakemap writes its output to stdout. Typical usage of xmakemap is to redirect output from xmakemap to a file (e.g., <code>\$HOME/.xmodmaprc</code>), edit this file to make necessary customizations, and then run xmodmap on the file (i.e., <code>\$HOME/.xmodmaprc</code>).</p> <p>For example:</p> <pre>xmakemap > \$HOME/.xmodmaprc [... make necessary customization to \$HOME/.xmodmaprc ...] xmodmap \$HOME/.xmodmaprc</pre> <p>If you would like your keytable to be customized each time OpenWindows is run, you should place the appropriate command in <code>\$HOME/.xinitrc</code> (copy <code>\$OPENWINHOME/lib/Xinitrc</code> to <code>\$HOME/.xinitrc</code> if it does not already exist).</p> <p>For example:</p> <pre>if [-f \$HOME/.xmodmaprc]; then xmodmap \$HOME/.xmodmaprc fi</pre> |
| NOTES | OpenWindows (i.e., an X server) must be running to use xmakemap . Read "Notes" in <code>\$HOME/.xmodmaprc</code> . |
| SEE ALSO | xmodmap (1) |

| NAME | xman – Manual page display program for the X Window System | | | | | | | | |
|-------------------------|---|-----------|--------------|-------|-------|------|-------------------|------|------------------|
| SYNOPSIS | xman [<i>-options . . .</i>] | | | | | | | | |
| DESCRIPTION | <p>Xman is a manual page browser. The default size of the initial xman window is small so that you can leave it running throughout your entire login session. In the initial window there are three options: <i>Help</i> will pop up a window with on-line help, <i>Quit</i> will exit, and <i>Manual Page</i> will pop up a window with a manual page browser in it. Typing Control-S will pop up a window prompting for a specific manual page to display. You may display more than one manual page browser window at a time from a single execution of xman.</p> <p>For further information on using xman, please read the on-line help information. Most of this manual will discuss customization of xman.</p> | | | | | | | | |
| OPTIONS | <p>Xman supports all standard Toolkit command line arguments (see X11(7)). The following additional arguments are supported.</p> <p>-helpfile <i>filename</i> Specifies a helpfile to use other than the default.</p> <p>-bothshown Allows both the manual page and manual directory to be on the screen at the same time.</p> <p>-notopbox Starts without the Top Menu with the three buttons in it.</p> <p>-geometry <i>WxH+X+Y</i> Sets the size and location of the Top Menu with the three buttons in it.</p> <p>-pagesize <i>WxH+X+Y</i> Sets the size and location of all the Manual Pages.</p> | | | | | | | | |
| CUSTOMIZING XMAN | <p>Xman allows customization of both the directories to be searched for manual pages, and the name that each directory will map to in the <i>Sections</i> menu. Xman determines which directories it will search by reading the <i>MANPATH</i> environment variable. If no <i>MANPATH</i> is found then the directory is <i>/usr/man</i> is searched on POSIX systems. This environment is expected to be a colon-separated list of directories for xman to search.</p> <p>setenv MANPATH /mit/kit/man:/usr/man</p> <p>By default, xman will search each of the following directories (in each of the directories specified in the users <i>MANPATH</i>) for manual pages. If manual pages exist in that directory then they are added to list of manual pages for the corresponding menu item. A menu item is only displayed for those sections that actually contain manual pages.</p> <table border="0" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;">Directory</th> <th style="text-align: left;">Section Name</th> </tr> <tr> <th style="text-align: left;">-----</th> <th style="text-align: left;">-----</th> </tr> </thead> <tbody> <tr> <td>man1</td> <td>(1) User Commands</td> </tr> <tr> <td>man2</td> <td>(2) System Calls</td> </tr> </tbody> </table> | Directory | Section Name | ----- | ----- | man1 | (1) User Commands | man2 | (2) System Calls |
| Directory | Section Name | | | | | | | | |
| ----- | ----- | | | | | | | | |
| man1 | (1) User Commands | | | | | | | | |
| man2 | (2) System Calls | | | | | | | | |

| | |
|------|-------------------------|
| man3 | (3) Subroutines |
| man4 | (4) Devices |
| man5 | (5) File Formats |
| man6 | (6) Games |
| man7 | (7) Miscellaneous |
| man8 | (8) Sys. Administration |
| manl | (l) Local |
| mann | (n) New |
| mano | (o) Old |

For instance, a user has three directories in her manual path and each contain a directory called *man3*. All these manual pages will appear alphabetically sorted when the user selects the menu item called (3) *Subroutines*. If there is no directory called *mano* in any of the directories in her MANPATH, or there are no manual pages in any of the directories called *mano* then no menu item will be displayed for the section called (o) *Old*.

THE MANDESC FILE

By using the *mandesc* file a user or system manager is able to more closely control which manual pages will appear in each of the sections represented by menu items in the *Sections* menu. This functionality is only available on a section by section basis, and individual manual pages may not be handled in this manner. (Although generous use of symbolic links — see **ln(1)** — will allow almost any configuration you can imagine.)

The format of the *mandesc* file is a character followed by a label. The character determines which of the sections will be added under this label. For instance suppose that you would like to create an extra menu item that contains all programmer subroutines. This label should contain all manual pages in both sections two and three. The *mandesc* file would look like this:

```
2Programmer Subroutines
3Programmer Subroutines
```

This will add a menu item to the *Sections* menu that would bring up a listing of all manual pages in sections two and three of the *Programmers Manual*. Since the label names are *exactly* the same they will be added to the same section. Note, however, that the original sections still exist.

If you want to completely ignore the default sections in a manual directory then add the line:

```
no default sections
```

anywhere in your *mandesc* file. This keeps **xman** from searching the default manual sections *In that directory only*. As an example, suppose you want to do the same thing as above, but you don't think that it is useful to have the *System Calls* or *Subroutines* sections any longer. You would need to duplicate the default entries, as well as adding your new one.

no default sections
 1(1) User Commands
 2Programmer Subroutines
 3Programmer Subroutines
 4(4) Devices
 5(5) File Formats
 6(6) Games
 7(7) Miscellaneous
 8(8) Sys. Administration
 l(l) Local
 n(n) New
 o(o) Old

Xman will read any section that is of the form *man<character>*, where <character> is an upper or lower case letter (they are treated distinctly) or a numeral (0-9). Be warned, however, that **man(1)** and **catman(1M)** will not search directories that are non-standard.

WIDGETS

In order to specify resources, it is useful to know the hierarchy of the widgets which compose **xman**. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

```
Xman xman    (This widget is never used)
  TopLevelShell topbox
    Form form
      Label topLabel
      Command helpButton
      Command quitButton
      Command manpageButton
    TransientShell search
      DialogWidgetClass dialog
        Label label
        Text value
        Command manualPage
        Command apropos
        Command cancel
    TransientShell pleaseStandBy
      Label label
  TopLevelShell manualBrowser
    Paned Manpage_Vpane
      Paned horizPane
        MenuButton options
        MenuButton sections
        Label manualBrowser
    Viewport directory
      List directory
```

```

List directory
.
. (one for each section,
. created on the fly)
.
ScrollByLine manualPage
SimpleMenu optionMenu
SmeBSB displayDirectory
SmeBSB displayManualPage
SmeBSB help
SmeBSB search
SmeBSB showBothScreens
SmeBSB removeThisManpage
SmeBSB openNewManpage
SmeBSB showVersion
SmeBSB quit
SimpleMenu sectionMenu
SmeBSB <name of section>
.
. (one for each section)
.
TransientShell search
DialogWidgetClass dialog
Label label
Text value
Command manualPage
Command apropos
Command cancel
TransientShell pleaseStandBy
Label label
TransientShell likeToSave
Dialog dialog
Label label
Text value
Command yes
Command no
TopLevelShell help
Paned Manpage_Vpane
Paned horizPane
MenuBar options
MenuBar sections
Label manualBrowser
ScrollByLine manualPage
SimpleMenu optionMenu
SmeBSB displayDirectory

```

SmeBSB displayManualPage
 SmeBSB help
 SmeBSB search
 SmeBSB showBothScreens
 SmeBSB removeThisManpage
 SmeBSB openNewManpage
 SmeBSB showVersion
 SmeBSB quit

APPLICATION RESOURCES

xman has the following application-specific resources which allow customizations unique to **xman**.

manualFontNormal (Class **Font**)

The font to use for normal text in the manual pages.

manualFontBold (Class **Font**)

The font to use for bold text in the manual pages.

manualFontItalic (Class **Font**)

The font to use for italic text in the manual pages.

directoryFontNormal (Class **Font**)

The font to use for the directory text.

bothShown (Class **Boolean**)

Either 'true' or 'false,' specifies whether or not you want both the directory and the manual page shown at start up.

directoryHeight (Class **DirectoryHeight**)

The height in pixels of the directory, when the directory and the manual page are shown simultaneously.

topCursor (Class **Cursor**)

The cursor to use in the top box.

helpCursor (Class **Cursor**)

The cursor to use in the help window.

manpageCursor (Class **Cursor**)

The cursor to use in the manual page window.

searchEntryCursor (Class **Cursor**)

The cursor to use in the search entry text widget.

pointerColor (Class **Foreground**)

This is the color of all the cursors (pointers) specified above. The name was chosen to be compatible with xterm.

helpFile (Class **File**)

Use this rather than the system default helpfile.

topBox (Class **Boolean**)

Either 'true' or 'false,' determines whether the top box (containing the help, quit and manual page buttons) or a manual page is put on

the screen at start-up. The default is true.

verticalList (Class **Boolean**)

Either 'true' or 'false,' determines whether the directory listing is vertically or horizontally organized. The default is horizontal (false).

**GLOBAL
ACTIONS**

Xman defines all user interaction through global actions. This allows the user to modify the translation table of any widget, and bind any event to the new user action. The list of actions supported by **xman** are:

GotoPage(*page*) When used in a manual page display window this will allow the user to move between a directory and manual page display. The *page* argument can be either **Directory** or **ManualPage**.

Quit() This action may be used anywhere, and will exit **xman**.

Search(*type, action*) Only useful when used in a search popup, this action will cause the search widget to perform the named search type on the string in the search popup's value widget. This action will also pop down the search widget. The *type* argument can be either **Apropos**, **Manpage** or **Cancel**. If an *action* of **Open** is specified then **xman** will open a new manual page to display the results of the search, otherwise **xman** will attempt to display the results in the parent of the search popup.

PopupHelp() This action may be used anywhere, and will popup the help widget.

PopupSearch() This action may be used anywhere except in a help window. It will cause the search popup to become active and visible on the screen, allowing the user search for a manual page.

CreateNewManpage() This action may be used anywhere, and will create a new manual page display window.

RemoveThisManpage() This action may be used in any manual page or help display window. When called it will remove the window, and clean up all resources associated with it.

SaveFormattedPage(*action*) This action can only be used in the **likeToSave** popup widget, and tells **xman** whether to **Save** or **Cancel** a save of the manual page that has just been formatted.

ShowVersion() This action may be called from any manual page or help display window, and will cause the informational display line to show the current version of **xman**.

FILES

<*manpath directory*>/man<*character*>

<*manpath directory*>/cat<*character*>

<*manpath directory*>/mandesc

`/usr/openwin/lib/app-defaults/Xman` specifies required resources
`/tmp` *Xman* creates temporary files in `/tmp` for all unformatted man pages and all apropos searches.

SEE ALSO **X11(7) man(1), apropos(1), catman(1M), Athena Widget Set**

ENVIRONMENT

DISPLAY the default host and display to use.

MANPATH the search path for manual pages. Directories are separated by colons (e.g. `/usr/man:/mit/kit/man:/foo/bar/man`).

XENVIRONMENT to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

XAPPLRESDIR A string that will have “Xman” appended to it. This string will be the full path name of a user app-defaults file to be merged into the resource database after the system app-defaults file, and before the resources that are attached to the display.

COPYRIGHT Copyright 1988 by Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHORS Chris Peterson, MIT X Consortium from the V10 version written by Barry Shein formerly of Boston University.

| | |
|---------------------|---|
| NAME | xmh – send and read mail with an X interface to MH |
| SYNOPSIS | xmh [<i>–path mailpath</i>] [<i>–initial foldername</i>] [<i>–flag</i>] [<i>–toolkitoption ...</i>] |
| DESCRIPTION | <p>The xmh program provides a graphical user interface to the <i>MH</i> Message Handling System. To actually do things with your mail, it makes calls to the <i>MH</i> package. Electronic mail messages may be composed, sent, received, replied to, forwarded, sorted, and stored in folders. xmh provides extensive mechanism for customization of the user interface.</p> <p>This document introduces many aspects of the Athena Widget Set.</p> |
| OPTIONS | <p>–path <i>directory</i></p> <p>This option specifies an alternate collection of mail folders in which to process mail. The directory is specified as an absolute pathname. The default mail path is the value of the Path component in the <i>MH</i> profile, which is determined by the MH environment variable and defaults to <code>\$HOME/.mh_profile</code>. <code>\$HOME/Mail</code> will be used as the path if the <i>MH</i> Path is not given in the profile.</p> <p>–initial <i>folder</i></p> <p>This option specifies an alternate folder which may receive new mail and is initially opened by xmh. The default initial folder is “inbox”.</p> <p>–flag</p> <p>This option will cause xmh to change the appearance of appropriate folder buttons and to request the window manager to change the appearance of the xmh icon when new mail has arrived. By default, xmh will change the appearance of the “inbox” folder button when new mail is waiting. The application-specific resource checkNewMail can be used to turn off this notification, and the –flag option will still override it.</p> <p>These three options have corresponding application-specific resources, MailPath, InitialFolder, and MailWaitingFlag, which can be specified in a resource file.</p> <p>The standard toolkit command line options are given in X11(7).</p> |
| INSTALLATION | <p>xmh requires that the user is already set up to use <i>MH</i>, version 6. To do so, see if there is a file called <code>.mh_profile</code> in your home directory. If it exists, check to see if it contains a line that starts with “Current-Folder”. If it does, you’ve been using version 4 or earlier of <i>MH</i>; to convert to version 6, you must remove that line. (Failure to do so causes spurious output to <code>stderr</code>, which can hang xmh depending on your setup.)</p> <p>If you do not already have a <code>.mh_profile</code>, you can create one (and everything else you need) by typing “inc” to the shell. You should do this before using xmh to incorporate new mail.</p> <p>For more information, refer to the mh(1) documentation.</p> <p>Much of the user interface of xmh is configured in the <i>Xmh</i> application class defaults file; if this file was not installed properly a warning message will appear when xmh is used. xmh is backwards compatible with the R4 application class defaults file.</p> |

The default value of the SendBreakWidth resource has changed since R4.

BASIC SCREEN LAYOUT

xmh starts out with a single window, divided into four major areas:

- Six buttons with pull-down command menus.
- A collection of buttons, one for each top level folder. New users of *MH* will have two folders, “drafts” and “inbox”.
- A listing, or Table of Contents, of the messages in the open folder. Initially, this will show the messages in “inbox”.
- A view of one of your messages. Initially this is blank.

XMH AND THE ATHENA WIDGET SET

xmh uses the X Toolkit Intrinsics and the Athena Widget Set. Many of the features described below (scrollbars, buttonboxes, etc.) are actually part of the Athena Widget Set, and are described here only for completeness. For more information, see the Athena Widget Set documentation.

SCROLLBARS

Some parts of the main window will have a vertical area on the left containing a grey bar. This area is a *scrollbar*. They are used whenever the data in a window takes up more space than can be displayed. The grey bar indicates what portion of your data is visible. Thus, if the entire length of the area is grey, then you are looking at all your data. If only the first half is grey, then you are looking at the top half of your data. The message viewing area will have a horizontal scrollbar if the text of the message is wider than the viewing area.

You can use the pointer in the scrollbar to change what part of the data is visible. If you click with pointer button 2, the top of the grey area will move to where the pointer is, and the corresponding portion of data will be displayed. If you hold down pointer button 2, you can drag around the grey area. This makes it easy to get to the top of the data: just press with button 2, drag off the top of the scrollbar, and release.

If you click with button 1, then the data to the right of the pointer will scroll to the top of the window. If you click with pointer button 3, then the data at the top of the window will scroll down to where the pointer is.

BUTTONBOXES, BUTTONS, AND MENUS

Any area containing many words or short phrases, each enclosed in a rectangular or rounded boundary, is called a *buttonbox*. Each rectangle or rounded area is actually a button that you can press by moving the pointer onto it and pressing pointer button 1. If a given buttonbox has more buttons in it than can fit, it will be displayed with a scrollbar, so you can always scroll to the button you want.

Some buttons have pull-down menus. Pressing the pointer button while the pointer is over one of these buttons will pull down a menu. Continuing to hold the button down while moving the pointer over the menu, called dragging the pointer, will highlight each

selectable item on the menu as the pointer passes over it. To select an item in the menu, release the pointer button while the item is highlighted.

ADJUSTING THE RELATIVE SIZES OF AREAS

If you're not satisfied with the sizes of the various areas of the main window, they can easily be changed. Near the right edge of the border between each region is a black box, called a *grip*. Simply point to that grip with the pointer, press a pointer button, drag up or down, and release. Exactly what happens depends on which pointer button you press.

If you drag with the pointer button 2, then only that border will move. This mode is simplest to understand, but is the least useful.

If you drag with pointer button 1, then you are adjusting the size of the window above. **xmh** will attempt to compensate by adjusting some window below it.

If you drag with pointer button 3, then you are adjusting the size of the window below. **xmh** will attempt to compensate by adjusting some window above it.

All windows have a minimum and maximum size; you will never be allowed to move a border past the point where it would make a window have an invalid size.

PROCESSING YOUR MAIL

This section will define the concepts of the selected folder, current folder, selected message(s), current message, selected sequence, and current sequence. Each **xmh** command is introduced.

For use in customization, action procedures corresponding to each command are given; these action procedures can be used to customize the user interface, particularly the keyboard accelerators and the functionality of the buttons in the optional button box created by the application resource **CommandButtonCount**.

FOLDERS AND SEQUENCES

A folder contains a collection of mail messages, or is empty. **xmh** supports folders with one level of subfolders.

The selected folder is whichever foldername appears in the bar above the folder buttons. Note that this is not necessarily the same folder that is currently being viewed. To change the selected folder, just press on the desired folder button with pointer button 1; if that folder has subfolders, select a folder from the pull-down menu.

The Table of Contents, or toc, lists the messages in the viewed folder. The title bar above the Table of Contents displays the name of the viewed folder.

The toc title bar also displays the name of the viewed sequence of messages within the viewed folder. Every folder has an implicit "all" sequence, which contains all the messages in the folder, and initially the toc title bar will show "inbox:all".

FOLDER COMMANDS

The *Folder* command menu contains commands of a global nature:

Open Folder

Display the data in the selected folder. Thus, the selected folder also becomes the viewed folder. The action procedure corresponding to this command is **XmhOpenFolder**(*[foldername]*). It takes an optional argument as the name of a folder to select and open; if no folder is specified, the selected folder is opened. It may be specified as part of an event translation from a folder menu button or from a folder menu, or as a binding of a keyboard accelerator to any widget other than the folder menu buttons or the folder menus.

Open Folder in New Window

Displays the selected folder in an additional main window. Note, however, that you cannot reliably display the same folder in more than one window at a time, although **xmh** will not prevent you from trying. The corresponding action is **XmhOpenFolderInNewWindow**().

Create Folder

Create a new folder. You will be prompted for a name for the new folder; to enter the name, move the pointer to the blank box provided and type. Subfolders are created by specifying the parent folder, a slash, and the subfolder name. For example, to create a folder named "xmh" which is a subfolder of an existing folder named "clients", type "clients/xmh". Click on the Okay button when finished, or just type Return; click on Cancel to cancel this operation. The action corresponding to Create Folder is **XmhCreateFolder**().

Delete Folder

Destroy the selected folder. You will be asked to confirm this action (see CONFIRMATION WINDOWS). Destroying a folder will also destroy any subfolders of that folder. The corresponding action is **XmhDeleteFolder**().

Close Window

Exits **xmh**, after first confirming that you won't lose any changes; or, if selected from any additional **xmh** window, simply closes that window. The corresponding action is **XmhClose**().

HIGHLIGHTED MESSAGES, SELECTED MESSAGES, CURRENT MESSAGE

It is possible to highlight a set of adjacent messages in the area of the Table of Contents. To highlight a message, click on it with pointer button 1. To highlight a range of messages, click on the first one with pointer button 1 and on the last one with pointer button 3; or press pointer button 1, drag, and release. To extend a range of selected messages, use pointer button 3. To highlight all messages in the table of contents, click rapidly three times with pointer button 1. To cancel any selection in the table of contents, click rapidly twice.

The selected messages are the same as the highlighted messages, if any. If no messages are highlighted, then the selected messages are considered the same as the current message.

The current message is indicated by a '+' next to the message number. It usually corresponds to the message currently being viewed. Upon opening a new folder, for

example, the current message will be different from the viewed message. When a message is viewed, the title bar above the view will identify the message.

TABLE OF CONTENTS COMMANDS

The *Table of Contents* command menu contains commands which operate on the open, or viewed, folder.

Incorporate New Mail

Add any new mail received to viewed folder, and set the current message to be the first new message. This command is selectable in the menu and will execute only if the viewed folder is allowed to receive new mail. By default, only "inbox" is allowed to incorporate new mail. The corresponding action is **XmhIncorporateNewMail()**.

Commit Changes Execute all deletions, moves, and copies that have been marked in this folder. The corresponding action is **XmhCommitChanges()**.

Pack Folder Renumber the messages in this folder so they start with 1 and increment by 1. The corresponding action is **XmhPackFolder()**.

Sort Folder Sort the messages in this folder in chronological order. (As a side effect, this may also pack the folder.) The corresponding action is **XmhSortFolder()**.

Rescan Folder Rebuild the list of messages. This can be used whenever you suspect that **xmh**'s idea of what messages you have is wrong. (In particular, this is necessary if you change things using straight *MH* commands without using **xmh**.) The corresponding action is **XmhForceRescan()**.

MESSAGE COMMANDS

The *Message* command menu contains commands which operate on the selected message(s), or if there are no selected messages, the current message.

Compose Message Composes a new message. A new window will be brought up for composition; a description of it is given in the COMPOSITION WINDOWS section below. This command does not affect the current message. The corresponding action is **XmhComposeMessage()**.

View Next Message

View the first selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message after the current message. The corresponding action is **XmhViewNextMessage()**.

View Previous

View the last selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message before the current message. The corresponding action is **XmhViewPrevious()**.

| | |
|---------------------------|--|
| Delete | Mark the selected messages for deletion. If no messages are highlighted, mark the current message for deletion and automatically display the next unmarked message. The corresponding action is XmhMarkDeleted() . |
| Move | Mark the selected messages to be moved into the currently selected folder. (If the selected folder is the same as the viewed folder, this command will just beep.) If no messages are highlighted, mark the current message to be moved and display the next unmarked message. The corresponding action is XmhMarkMove() . |
| Copy as Link | Mark the selected messages to be copied into the selected folder. (If the selected folder is the same as the viewed folder, this command will just beep.) If no messages are highlighted, mark the current message to be copied. Note that messages are actually linked, not copied; editing a message copied by xmh will affect all copies of the message. The corresponding action is XmhMarkCopy() . |
| Unmark | Remove any of the above three marks from the selected messages, or the current message, if none are highlighted. The corresponding action is XmhUnmark() . |
| View in New | Create a new window containing only a view of the first selected message, or the current message, if none are highlighted. The corresponding action is XmhViewInNewWindow() . |
| Reply | Create a composition window in reply to the first selected message, or the current message, if none are highlighted. The corresponding action is XmhReply() . |
| Forward | Create a composition window whose body is initialized to contain an encapsulation of of the selected messages, or the current message if none are highlighted. The corresponding action is XmhForward() . |
| Use as Composition | Create a composition window whose body is initialized to be the contents of the first selected message, or the current message if none are selected. Any changes you make in the composition will be saved in a new message in the “drafts” folder, and will not change the original message. However, there is an exception to this rule. If the message to be used as composition was selected from the “drafts” folder, (see BUGS), the changes will be reflected in the original message (see COMPOSITION WINDOWS). The action procedure corresponding to this command is XmhUseAsComposition() . |
| Print | Print the selected messages, or the current message if none are selected. xmh normally prints by invoking the enscript command, but this can be customized with the xmh application-specific resource PrintCommand . The corresponding action is XmhPrint() . |

**SEQUENCE
COMMANDS**

The *Sequence* command menu contains commands pertaining to message sequences (See MESSAGE-SEQUENCES), and a list of the message-sequences defined for the currently viewed folder. The selected message-sequence is indicated by a check mark in its entry in the margin of the menu. To change the selected message-sequence, select a new message-sequence from the sequence menu.

Pick Messages Define a new message-sequence. The corresponding action is **XmhPickMessages()**.

The following menu entries will be sensitive only if the current folder has any message-sequences other than the “all” message-sequence.

Open Sequence Change the viewed sequence to be the same as the selected sequence. The corresponding action is **XmhOpenSequence()**.

Add to Sequence Add the selected messages to the selected sequence. The corresponding action is **XmhAddToSequence()**.

Remove from Sequence

Remove the selected messages from the selected sequence. The corresponding action is **XmhRemoveFromSequence()**.

Delete Sequence Remove the selected sequence entirely. The messages themselves are not affected; they simply are no longer grouped together to define a message-sequence. The corresponding action is **XmhDeleteSequence()**.

VIEW COMMANDS

Commands in the *View* menu and in the buttonboxes of view windows (which result from the *Message* menu command **View In New**) correspond in functionality to commands of the same name in the *Message* menu, but they operate on the viewed message rather than the selected messages or current message.

Close Window When the viewed message is in a separate view window, this command will close the view, after confirming the status of any unsaved edits. The corresponding action procedure is **XmhCloseView()**.

Reply Create a composition window in reply to the viewed message. The related action procedure is **XmhViewReply()**.

Forward Create a composition window whose body is initialized contain an encapsulation of the viewed message. The corresponding action is **XmhViewForward()**.

Use As Composition

Create a composition window whose body is initialized to be the contents of the viewed message. Any changes made in the composition window will be saved in a new message in the “drafts” folder, and will not change the original message. An exception: if the viewed message was selected from the “drafts” folder, (see BUGS)

| | |
|---------------------|--|
| | the original message is edited. The action procedure corresponding to this command is XmhViewUseAsComposition() . |
| Edit Message | This command enables the direct editing of the viewed message. The action procedure is XmhEditView() . |
| Save Message | This command is insensitive until the message has been edited; when activated, edits will be saved to the original message in the view. The corresponding action is XmhSaveView() . |
| Print | Print the viewed message. xmh prints by invoking the enscript command, but this can be customized with the application-specific resource PrintCommand . The corresponding action procedure is XmhPrintView() . |
| Delete | Marks the viewed message for deletion. The corresponding action procedure is XmhViewMarkDelete() . |

OPTIONS

The *Options* menu contains one entry.

Read in Reverse

When selected, a check mark appears in the margin of this menu entry. Read in Reverse will switch the meaning of the next and previous messages, and will increment to the current message marker in the opposite direction. This is useful if you want to read your messages in the order of most recent first. The option acts as a toggle; select it from the menu a second time to undo the effect. The check mark appears when the option is selected.

COMPOSITION WINDOWS

Composition windows are created by selecting **Compose Message** from the *Message* command menu, or by selecting **Reply** or **Forward** or **Use as Composition** from the *Message* or *View* command menu. These are used to compose mail messages. Aside from the normal text editing functions, there are six command buttons associated with composition windows:

| | |
|------------------------|---|
| Close Window | Close this composition window. If changes have been made since the most recent Save or Send, you will be asked to confirm losing them. The corresponding action is XmhCloseView() . |
| Send | Send this composition. The corresponding action is XmhSend() . |
| New Headers | Replace the current composition with an empty message. If changes have been made since the most recent Send or Save, you will be asked to confirm losing them. The corresponding action is XmhResetCompose() . |
| Compose Message | Bring up another new composition window. The corresponding action is XmhComposeMessage() . |

- Save Message** Save this composition in your drafts folder. Then you can safely close the composition. At some future date, you can continue working on the composition by opening the drafts folder, selecting the message, and using the “Use as Composition” command. The corresponding action is **XmhSave()**.
- Insert** Insert a related message into the composition. If the composition window was created with a “Reply” command, the related message is the message being replied to, otherwise no related message is defined and this button is insensitive. The message may be filtered before being inserted; see **ReplyInsertFilter** under APPLICATION RESOURCES for more information. The corresponding action is **XmhInsert()**.

ACCELERATORS

Accelerators are shortcuts. They allow you to invoke commands without using the menus, either from the keyboard or by using the pointer.

xmh defines pointer accelerators for common actions: To select and view a message with a single click, use pointer button 2 on the message’s entry in the table of contents. To select and open a folder or a sequence in a single action, make the folder or sequence selection with pointer button 2.

To mark the highlighted messages, or current message if none have been highlighted, to be moved to a folder in a single action, use pointer button 3 to select the target folder and simultaneously mark the messages. Similarly, selecting a sequence with pointer button 3 will add the highlighted or current message(s) to that sequence. In both of these operations, the selected folder or sequence and the viewed folder or sequence are not changed.

xmh defines the following keyboard accelerators over the surface of the main window, except in the view area while editing a message:

| | |
|------------|--|
| Meta-I | Incorporate New Mail |
| Meta-C | Commit Changes |
| Meta-R | Rescan Folder |
| Meta-P | Pack Folder |
| Meta-S | Sort Folder |
| Meta-space | View Next Message |
| Meta-c | Mark Copy |
| Meta-d | Mark Deleted |
| Meta-f | Forward the selected or current message |
| Meta-m | Mark Move |
| Meta-n | View Next Message |
| Meta-p | View Previous Message |
| Meta-r | Reply to the selected or current message |
| Meta-u | Unmark |

| | |
|--------|---------------------------------------|
| Ctrl-V | Scroll the table of contents forward |
| Meta-V | Scroll the table of contents backward |
| Ctrl-v | Scroll the view forward |
| Meta-v | Scroll the view backward |

TEXT EDITING COMMANDS

All of the text editing commands are actually defined by the Text widget in the Athena Widget Set. The commands may be bound to different keys than the defaults described below through the X Toolkit Intrinsic key re-binding mechanisms. See the X Toolkit Intrinsic and the Athena Widget Set documentation for more details.

Whenever you are asked to enter any text, you will be using a standard text editing interface. Various control and meta keystroke combinations are bound to a somewhat Emacs-like set of commands. In addition, the pointer buttons may be used to select a portion of text or to move the insertion point in the text. Pressing pointer button 1 causes the insertion point to move to the pointer. Double-clicking button 1 selects a word, triple-clicking selects a line, quadruple-clicking selects a paragraph, and clicking rapidly five times selects everything. Any selection may be extended in either direction by using pointer button 3.

In the following, a *line* refers to one displayed row of characters in the window. A *paragraph* refers to the text between carriage returns. Text within a paragraph is broken into lines for display based on the current width of the window. When a message is sent, text is broken into lines based upon the values of the **SendBreakWidth** and **SendWidth** application-specific resources.

The following keystroke combinations are defined:

| | | | |
|--------|---------------------------|--------|--------------------------|
| Ctrl-a | Beginning Of Line | Meta-b | Backward Word |
| Ctrl-b | Backward Character | Meta-f | Forward Word |
| Ctrl-d | Delete Next Character | Meta-i | Insert File |
| Ctrl-e | End Of Line | Meta-k | Kill To End Of Paragraph |
| Ctrl-f | Forward Character | Meta-q | Form Paragraph |
| Ctrl-g | Multiply Reset | Meta-v | Previous Page |
| Ctrl-h | Delete Previous Character | Meta-y | Insert Current Selection |
| Ctrl-j | Newline And Indent | Meta-z | Scroll One Line Down |
| Ctrl-k | Kill To End Of Line | Meta-d | Delete Next Word |
| Ctrl-l | Redraw Display | Meta-D | Kill Word |
| Ctrl-m | Newline | Meta-h | Delete Previous Word |
| Ctrl-n | Next Line | Meta-H | Backward Kill Word |
| Ctrl-o | Newline And Backup | Meta-< | Beginning Of File |
| Ctrl-p | Previous Line | Meta-> | End Of File |
| Ctrl-r | Search/Replace Backward | Meta-] | Forward Paragraph |
| Ctrl-s | Search/Replace Forward | Meta-[| Backward Paragraph |
| Ctrl-t | Transpose Characters | | |

| | | | |
|--------|--------------------|----------------------|----------------------|
| Ctrl-u | Multiply by 4 | Meta-Delete | Delete Previous Word |
| Ctrl-v | Next Page | Meta-Shift Delete | Kill Previous Word |
| Ctrl-w | Kill Selection | Meta-Backspace | Delete Previous Word |
| Ctrl-y | Unkill | Meta-Shift Backspace | Kill Previous Word |
| Ctrl-z | Scroll One Line Up | | |

In addition, the pointer may be used to copy and paste text:

| | |
|-----------------|----------------------------------|
| Button 1 Down | Start Selection |
| Button 1 Motion | Adjust Selection |
| Button 1 Up | End Selection (copy) |
| Button 2 Down | Insert Current Selection (paste) |
| Button 3 Down | Extend Current Selection |
| Button 3 Motion | Adjust Selection |
| Button 3 Up | End Selection (copy) |

CONFIRMATION DIALOG BOXES

Whenever you press a button that may cause you to lose some work or is otherwise dangerous, a popup dialog box will appear asking you to confirm the action. This window will contain an "Abort" or "No" button and a "Confirm" or "Yes" button. Pressing the "No" button cancels the operation, and pressing the "Yes" will proceed with the operation.

Some dialog boxes contain messages from *MH*. Occasionally when the message is more than one line long, not all of the text will be visible. Clicking on the message field will cause the dialog box to resize so that you can read the entire message.

MESSAGE- SEQUENCES

An *MH* message sequence is just a set of messages associated with some name. They are local to a particular folder; two different folders can have sequences with the same name. The sequence named "all" is predefined in every folder; it consists of the set of all messages in that folder. As many as nine sequences may be defined for each folder, including the predefined "all" sequence. (The sequence "cur" is also usually defined for every folder; it consists of only the current message. **xmh** hides "cur" from the user, instead placing a "+" by the current message. Also, **xmh** does not support *MH*'s "unseen" sequence, so that one is also hidden from the user.)

The message sequences for a folder (including one for "all") are displayed in the "Sequence" menu, below the sequence commands. The table of contents (also known as the "toc") is at any one time displaying one message sequence. This is called the "viewed sequence", and its name will be displayed in the toc title bar after the folder name. Also, at any time one of the sequences in the menu will have a check mark next to it. This is called the "selected sequence". Note that the viewed sequence and the selected sequence are not necessarily the same. (This all pretty much corresponds to the way folders work.)

The **Open Sequence**, **Add to Sequence**, **Remove from Sequence**, and **Delete Sequence** commands are active only if the viewed folder contains message-sequences other than “all” sequence.

Note that none of the above actually affect whether a message is in the folder. Remember that a sequence is a set of messages within the folder; the above operations just affect what messages are in that set.

To create a new sequence, select the “Pick” menu entry. A new window will appear, with lots of places to enter text. Basically, you can describe the sequence’s initial set of messages based on characteristics of the message. Thus, you can define a sequence to be all the messages that were from a particular person, or with a particular subject, and so on. You can also connect things up with boolean operators, so you can select all things from “weissman” with a subject containing “xmh”.

The layout should be fairly obvious. The simplest cases are the easiest: just point to the proper field and type. If you enter in more than one field, it will only select messages which match all non-empty fields.

The more complicated cases arise when you want things that match one field or another one, but not necessarily both. That’s what all the “or” buttons are for. If you want all things with subjects that include “xmh” or “xterm”, just press the “or” button next to the “Subject:” field. Another box will appear where you can enter another subject.

If you want all things either from “weissman” or with subject “xmh”, but not necessarily both, select the “-Or-” button. This will essentially double the size of the form. You can then enter “weissman” in a from: box on the top half, and “xmh” in a subject: box on the lower part.

If you select the “Skip” button, then only those messages that *don’t* match the fields on that row are included.

Finally, in the bottom part of the window will appear several more boxes. One is the name of the sequence you’re defining. (It defaults to the name of the selected sequence when “Pick” was pressed, or to “temp” if “all” was the selected sequence.) Another box defines which sequence to look through for potential members of this sequence; it defaults to the viewed sequence when “Pick” was pressed.

Two more boxes define a date range; only messages within that date range will be considered. These dates must be entered in RFC 822-style format: each date is of the form “dd mmm yy hh:mm:ss zzz”, where dd is a one or two digit day of the month, mmm is the three-letter abbreviation for a month, and yy is a year. The remaining fields are optional: hh, mm, and ss specify a time of day, and zzz selects a time zone. Note that if the time is left out, it defaults to midnight; thus if you select a range of “7 nov 86” – “8 nov 86”, you will only get messages from the 7th, as all messages on the 8th will have arrived after midnight.

“Date field” specifies which field in the header to look at for this date range; it defaults to “Date”. If the sequence you’re defining already exists, you can optionally merge the old set with the new; that’s what the “Yes” and “No” buttons are all about. Finally, you can “OK” the whole thing, or “Cancel” it.

In general, most people will rarely use these features. However, it’s nice to occasionally use “Pick” to find some messages, look through them, and then hit “Delete Sequence” to put things back in their original state.

WIDGET HIERARCHY

In order to specify resources, it is useful to know the hierarchy of widgets which compose **xmh**. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name. The application class name is **Xmh**.

The hierarchy of the main toc and view window is identical for additional toc and view windows, except that a **TopLevelShell** widget is inserted in the hierarchy between the application shell and the **Paned** widget.

Xmh xmh

 Paned xmh

 SimpleMenu folderMenu

 SmeBSB open

 SmeBSB openInNew

 SmeBSB create

 SmeBSB delete

 SmeLine line

 SmeBSB close

 SimpleMenu tocMenu

 SmeBSB inc

 SmeBSB commit

 SmeBSB pack

 SmeBSB sort

 SmeBSB rescan

 SimpleMenu messageMenu

 SmeBSB compose

 SmeBSB next

 SmeBSB prev

 SmeBSB delete

 SmeBSB move

 SmeBSB copy

 SmeBSB unmark

 SmeBSB viewNew

 SmeBSB reply

 SmeBSB forward

 SmeBSB useAsComp

```

    SmeBSB print
SimpleMenu sequenceMenu
    SmeBSB pick
    SmeBSB openSeq
    SmeBSB addToSeq
    SmeBSB removeFromSeq
    SmeBSB deleteSeq
    SmeLine line
    SmeBSB all
SimpleMenu viewMenu
    SmeBSB reply
    SmeBSB forward
    SmeBSB useAsComp
    SmeBSB edit
    SmeBSB save
    SmeBSB print
SimpleMenu optionMenu
    SmeBSB reverse
Viewport.Core menuBox.clip
    Box menuBox
        MenuButton folderButton
        MenuButton tocButton
        MenuButton messageButton
        MenuButton sequenceButton
        MenuButton viewButton
        MenuButton optionButton

Grip grip
Label folderTitlebar
Grip grip
Viewport.Core folders.clip
    Box folders
        MenuButton inbox
        MenuButton drafts
            SimpleMenu menu
                SmeBSB <folder_name>
                .
                .
                .

Grip grip
Label tocTitlebar
Grip grip
Text toc
    Scrollbar vScrollbar
Grip grip

```

```

Label viewTitlebar
Grip grip
Text view
    Scrollbar vScrollbar
    Scrollbar hScrollbar

```

The hierarchy of the Create Folder popup dialog box:

```

TransientShell prompt
  Dialog dialog
    Label label
    Text value
    Command okay
    Command cancel

```

The hierarchy of the Notice dialog box, which reports messages from MH:

```

TransientShell notice
  Dialog dialog
    Label label
    Text value
    Command confirm

```

The hierarchy of the Confirmation dialog box:

```

TransientShell confirm
  Dialog dialog
    Label label
    Command yes
    Command no

```

The hierarchy of the dialog box which reports errors:

```

TransientShell error
  Dialog dialog
    Label label
    Command OK

```

The hierarchy of the composition window:

```

TopLevelShell xmh
  Paned xmh
    Label composeTitlebar
    Text comp
    Viewport.Core compButtons.clip

```

```

Box compButtons
  Command close
  Command send
  Command reset
  Command compose
  Command save
  Command insert

```

The hierarchy of the view window:

```

TopLevelShell xmh
  Paned xmh
    Label viewTitlebar
    Text view
    Viewport.Core viewButtons.clip
      Box viewButtons
        Command close
        Command reply
        Command forward
        Command useAsComp
        Command edit
        Command save
        Command print
        Command delete

```

*The hierarchy of the pick window:
(Unnamed widgets have no name.)*

```

TopLevelShell xmh
  Paned xmh
    Label pickTitlebar
    Viewport.Core pick.clip
      Form form
        Form groupform

```

The first 6 rows of the pick window have identical structure:

```

Form rowform
  Toggle
  Toggle
  Label
  Text
  Command

```

```

Form rowform
  Toggle
  Toggle

```

```

Text
Text
Command
Form rowform
Command
Viewport.core pick.clip
Form form
From groupform
Form rowform
Label
Text
Label
Text
Form rowform
Label
Text
Label
Text
Label
Text
Form rowform
Label
Toggle
Toggle
Form rowform
Command
Command

```

APPLICATION- SPECIFIC RESOURCES

The application class name is **Xmh**. Application-specific resources are listed below by name. Application-specific resource class names always begin with an upper case character, but unless noted, are otherwise identical to the instance names given below.

Any of these options may also be specified on the command line by using the X Toolkit Intrinsics resource specification mechanism. Thus, to run **xmh** showing all message headers,

```
% xmh -xrm '*HideBoringHeaders:off'
```

If **TocGeometry**, **ViewGeometry**, **CompGeometry**, or **PickGeometry** are not specified, then the value of **Geometry** is used instead. If the resulting height is not specified (e.g., "", "=500", "+0-0"), then the default height of windows is calculated from fonts and line counts. If the width is not specified (e.g., "", "=x300", "-0+0"), then half of the display width is used. If unspecified, the height of a pick window defaults to half the height of the display.

The following resources are defined:

banner A short string that is the default label of the folder, Table of Contents, and view. The default is "xmh MIT X Consortium R5".

blockEventsOnBusy

Whether to disallow user input and show a busy cursor while **xmh** is busy processing a command. Default is true.

busyCursor

The name of the symbol used to represent the position of the pointer, displayed if **blockEventsOnBusy** is true, when **xmh** is processing a time-consuming command. The default is "watch".

busyPointerColor

The foreground color of the busy cursor. Default is XtDefaultForeground.

checkFrequency

How often to check for new mail, make checkpoints, and rescan the Table of Contents, in minutes. If **checkNewMail** is true, **xmh** checks to see if you have new mail each interval. If **makeCheckpoints** is true, checkpoints are made every fifth interval. Also every fifth interval, the Table of Contents is checked for inconsistencies with the file system, and rescanned if out of date. To prevent all of these checks from occurring, set **CheckFrequency** to 0. The default is 1. This resource is retained for backward compatibility with user resource files; see also **checkpointInterval**, **mailInterval**, and **rescanInterval**.

checkNewMail

If true, **xmh** will check at regular intervals to see if new mail has arrived for any of the top level folders and any opened subfolders. A visual indication will be given if new mail is waiting to be incorporated into a top level folder. Default is true. The interval can be adjusted with **mailInterval**.

checkpointInterval (class **Interval**)

Specifies in minutes how often to make checkpoints of volatile state, if **makeCheckpoints** is true. The default is 5 times the value of **checkFrequency**.

checkpointNameFormat

Specifies how checkpointed files are to be named. The value of this resource will be used to compose a file name by inserting the message number as a string in place of the required single occurrence of '%d'. If the value of the resource is the empty string, or if no '%d' occurs in the string, or if "%d" is the value of the resource, the default will be used instead. The default is "%d.CKP". Checkpointing is done in the folder of origin unless an absolute pathname is given. **xmh** does not assist the user in recovering checkpoints, nor does it provide for removal of the checkpoint files.

commandButtonCount

The number of command buttons to create in a button box in between the toc and the view areas of the main window. **xmh** will create these buttons with the names *button1*, *button2* and so on, in a box with the name *commandBox*. The default is 0. **xmh** users can specify labels and actions for the buttons in a private

resource file; see the section ACTIONS AND INTERFACE CUSTOMIZATION.

compGeometry

Initial geometry for windows containing compositions.

cursor The name of the symbol used to represent the pointer. Default is “left_ptr”.

debug Whether or not to print information to stderr as **xmh** runs. Default is false.

draftsFolder

The folder used for message drafts. Default is “drafts”.

geometry

Default geometry to use. Default is none.

hideBoringHeaders

If “on”, then **xmh** will attempt to skip uninteresting header lines within messages by scrolling them off the top of the view. Default is “on”.

initialFolder

Which folder to display on startup. May also be set with the command-line option **-initial**. Default is “inbox”.

initialIncFile

The absolute path name of your incoming mail drop file. In some installations, for example those using the Post Office Protocol, no file is appropriate. In this case, **initialIncFile** should not be specified, or may be specified as the empty string, and *inc* will be invoked without a **-file** argument. By default, this resource has no value. This resource is ignored if **xmh** finds an *.xmhcheck* file; see the section on multiple mail drops.

mailInterval (class **Interval**)

Specifies the interval in minutes at which the mail should be checked, if **mailWaitingFlag** or **checkNewMail** is true. The default is the value of **checkFrequency**.

mailPath

The full path prefix for locating your mail folders. May also be set with the command line option, **-path**. The default is the Path component in the *MH* profile, or “\$HOME/Mail” if none.

mailWaitingFlag

If true, **xmh** will attempt to set an indication in its icon when new mail is waiting to be retrieved. If **mailWaitingFlag** is true, then **checkNewMail** is assumed to be true as well. The **-flag** command line option is a quick way to turn on this resource.

makeCheckpoints

If true, **xmh** will attempt to save checkpoints of volatile edits. The default is false. The frequency of checkpointing is controlled by the resource **checkpointInterval**. For the location of checkpointing, see **checkpointNameFormat**.

- mhPath** What directory in which to find the *MH* commands. If a command isn't found in the user's path, then the path specified here is used. Default is `"/usr/local/mh6"`.
- newMailBitmap** (class **NewMailBitmap**)
The bitmap to show in the folder button when a folder has new mail. The default is `"black6"`.
- newMailIconBitmap** (class **NewMailBitmap**)
The bitmap suggested to the window manager for the icon when any folder has new mail. The default is `"flagup"`.
- noMailBitmap** (class **NoMailBitmap**)
The bitmap to show in the folder button when a folder has no new mail. The default is `"box6"`.
- noMailIconBitmap** (class **NoMailBitmap**)
The bitmap suggested to the window manager for the icon when no folders have new mail. The default is `"flagdown"`.
- pickGeometry**
Initial geometry for pick windows.
- pointerColor**
The foreground color of the pointer. Default is `XtDefaultForeground`.
- prefixWmAndIconName**
Whether to prefix the window and icon name with `"xmh: "`. Default is true.
- printCommand**
An *sh* command to execute to print a message. Note that `stdout` and `stderr` must be specifically redirected. If a message or range of messages is selected for printing, the full file paths of each message file are appended to the specified print command. The default is `"enscript >/dev/null 2>/dev/null"`.
- replyInsertFilter**
An *sh* command to be executed when the *Insert* button is activated in a composition window. The full path and filename of the source message is appended to the command before being passed to **sh**(1). The default filter is *cat*; i.e. it inserts the entire message into the composition. Interesting filters are: `sed 's/>/>/'` or `awk -e '{print " " $0}'` or `<mh directory>/lib/mhl -form mhl.body`.
- rescanInterval** (class **Interval**)
How often to check the Table of Contents of currently viewed folders and of folders with messages currently being viewed, and to update the Table of Contents if **xmh** sees inconsistencies with the file system in these folders. The default is 5 times the value of **checkFrequency**.
- reverseReadOrder**
When true, the next message will be the message prior to the current message in the table of contents, and the previous message will be the message after the current message in the table of contents. The default is false.

sendBreakWidth

When a message is sent from **xmh**, lines longer than this value will be split into multiple lines, each of which is no longer than **SendWidth**. This value may be overridden for a single message by inserting an additional line in the message header of the form *SendBreakWidth: value*. This line will be removed from the header before the message is sent. The default is 2000 (to allow for sending mail containing source patches).

sendWidth

When a message is sent from **xmh**, lines longer than **SendBreakWidth** characters will be split into multiple lines, each of which is no longer than this value. This value may be overridden for a single message by inserting an additional line in the message header of the form *SendWidth: value*. This line will be removed from the header before the message is sent. The default is 72.

showOnInc

Whether to automatically show the current message after incorporating new mail. Default is true.

skipCopied

Whether to skip over messages marked for copying when using “View Next Message” and “View Previous Message”. Default is true.

skipDeleted

Whether to skip over messages marked for deletion when using “View Next Message” and “View Previous Message”. Default is true.

skipMoved

Whether to skip over messages marked for moving to other folders when using “View Next Message” and “View Previous Message”. Default is true.

stickyMenu

If true, when popup command menus are used, the most recently selected entry will be under the cursor when the menu pops up. Default is false. See the file *clients/xmh/Xmh.sample* for an example of how to specify resources for popup command menus.

tempDir

Directory for **xmh** to store temporary files. For privacy, a user might want to change this to a private directory. Default is “/tmp”.

tocGeometry

Initial geometry for main **xmh** toc and view windows.

tocPercentage

The percentage of the main window that is used to display the Table of Contents. Default is 33.

tocWidth

How many characters to generate for each message in a folder’s table of contents. Default is 100. Use less if the geometry of the main **xmh** window results in the listing being clipped at the right hand boundary, or if you plan to use *mhl*

a lot, because it will be faster, and the extra characters may not be useful.

viewGeometry

Initial geometry for windows showing a view of a message.

MULTIPLE MAIL DROPS

Users may need to incorporate mail from multiple spool files or mail drops. If incoming mail is forwarded to the *MH slocal* program, it can be sorted as specified by the user into multiple incoming mail drops. Refer to the *MH* man page for *slocal* to learn how to specify forwarding and the automatic sorting of incoming mail in a *.maildelivery* file.

To inform **xmh** about the various mail drops, create a file in your home directory called *.xmhcheck*. In this file, a mapping between existing folder names and mail drops is created by giving a folder name followed by the absolute pathname of the mail drop site, with some white space separating them, one mapping per line. **xmh** will read this file whether or not resources are set for notification of new mail arrival, and will allow incorporation of new mail into any folder with a mail drop. **xmh** will invoke *inc* with the *-file* argument, and if **xmh** has been requested to check for new mail, it will check directly, instead of using *msgchk*.

An example of *.xmhcheck* file format, for the folders “inbox” and “xpert”:

```
inbox /var/mail/converse
xpert /users/converse/maildrops/xpert
```

ACTIONS AND INTERFACE CUSTOMIZATION

Because **xmh** provides action procedures which correspond to command functionality and installs accelerators, users can customize accelerators and new button functionality in a private resource file. For examples of specifying customized resources, see the file *mit/clients/xmh/Xmh.sample*. To understand the syntax, see the Appendix of the *X Toolkit Intrinsics* specification on *Translation Table Syntax*, and any general explanation of using and specifying *X* resources. Unpredictable results can occur if actions are bound to events or widgets for which they were not designed.

Here’s an example of how to bind actions to your own **xmh** buttons, and how to redefine the default accelerators so that the Meta key is not required, in case you don’t have access to the sample file mentioned above.

! To create buttons in the middle of the main window and give them semantics:

```
Xmh*CommandButtonCount:          5

Xmh*commandBox.button1.label:      Inc
Xmh*commandBox.button1.translations: #override\
    <Btn1Down>,<Btn1Up>: XmhIncorporateNewMail() unset()

Xmh*commandBox.button2.label:      Compose
Xmh*commandBox.button2.translations: #override\
    <Btn1Down>,<Btn1Up>: XmhComposeMessage() unset()
```

```
Xmh*commandBox.button3.label:      Next
Xmh*commandBox.button3.translations: #override\
    <Btn1Down>,<Btn1Up>: XmhViewNextMessage() unset()
```

```
Xmh*commandBox.button4.label:      Delete
Xmh*commandBox.button4.translations: #override\
    <Btn1Down>,<Btn1Up>: XmhMarkDelete() unset()
```

```
Xmh*commandBox.button5.label:      Commit
Xmh*commandBox.button5.translations: #override\
    <Btn1Down>,<Btn1Up>: XmhCommitChanges() unset()
```

! To redefine the accelerator bindings to exclude modifier keys,
! and add your own keyboard accelerator for Compose Message:

```
Xmh*tocMenu.accelerators: #override\n\
    !:<Key>I:      XmhIncorporateNewMail()\n\
    !:<Key>C:      XmhCommitChanges()\n\
    !:<Key>R:      XmhForceRescan()\n\
    !:<Key>P:      XmhPackFolder()\n\
    !:<Key>S:      XmhSortFolder()\n
Xmh*messageMenu.accelerators: #override\n\
    !:<Key>E:      XmhComposeMessage()\n\
    !:<Key>space:  XmhViewNextMessage()\n\
    !:<Key>c:      XmhMarkCopy()\n\
    !:<Key>d:      XmhMarkDelete()\n\
    !:<Key>f:      XmhForward()\n\
    !:<Key>m:      XmhMarkMove()\n\
    !:<Key>n:      XmhViewNextMessage()\n\
    !:<Key>p:      XmhViewPreviousMessage()\n\
    !:<Key>r:      XmhReply()\n\
    !:<Key>u:      XmhUnmark()\n
```

xmh provides action procedures which correspond to entries in the command menus; these are given in the sections describing menu commands, not here. In addition to the actions corresponding to commands in the menus, these action routines are defined:

XmhPushFolder(*[foldername, ...]*)

This action pushes each of its argument(s) onto a stack of foldernames. If no arguments are given, the selected folder is pushed onto the stack.

XmhPopFolder()

This action pops one foldername from the stack and sets the selected folder.

XmhPopupFolderMenu()

This action should always be taken when the user selects a folder button. A folder button represents a folder and zero or more subfolders. The menu of subfolders is built upon the first reference, by this routine. If there are no

subfolders, this routine will mark the folder as having no subfolders, and no menu will be built. In that case the menu button emulates a toggle button. When subfolders exist, the menu will popup, using the menu button action `PopupMenu()`.

XmhSetCurrentFolder()

This action allows menu buttons to emulate toggle buttons in the function of selecting a folder. This action is for menu button widgets only, and sets the selected folder.

XmhLeaveFolderButton()

This action ensures that the menu button behaves properly when the user moves the pointer out of the menu button window.

XmhPushSequence([sequencename, ...])

This action pushes each of its arguments onto the stack of sequence names. If no arguments are given, the selected sequence is pushed onto the stack.

XmhPopSequence()

This action pops one sequence name from the stack of sequence names, which then becomes the selected sequence.

XmhPromptOkayAction()

This action is equivalent to pressing the okay button in the Create Folder popup.

XmhReloadSeqLists()

This action rescans the contents of the public *MH* sequences for the currently opened folder and updates the sequence menu if necessary.

XmhShellCommand(parameter [, parameter])

At least one parameter must be specified. The parameters will be concatenated with a space character separator, into a single string, and the list of selected messages, or if no messages are selected, the current message, will be appended to the string of parameters. The string will be executed as a shell command. The messages are always given as absolute pathnames. It is an error to cause this action to execute when there are no selected messages and no current message.

XmhCheckForNewMail()

This action will check all mail drops known to *xmh*. If no mail drops have been specified by the user either through the *.xmhcheck* file or by the **initialIncFile** resource, the *MH* command *msgchk* is used to check for new mail, otherwise, *xmh* checks directly.

XmhWMProtocols([wm_delete_window] [wm_save_yourself])

This action is responsible for participation in window manager communication protocols. It responds to delete window and save yourself messages. The user can cause *xmh* to respond to one or both of these protocols, exactly as if the window manager had made the request, by invoking the action with the appropriate parameters. The action is insensitive to the case of the string parameters. If the event received is a `ClientMessage` event and parameters are

present, at least one of the parameters must correspond to the protocol requested by the event for the request to be honored by **xmh**.

CUSTOMIZATION USING *MH*

The initial text displayed in a composition window is generated by executing the corresponding *MH* command; i.e. *comp*, *repl*, or *forw*, and therefore message components may be customized as specified for those commands. *comp* is executed only once per invocation of **xmh** and the message template is re-used for every successive new composition.

xmh uses *MH* commands, including *inc*, *msgchk*, *comp*, *send*, *repl*, *forw*, *refile*, *rmm*, *pick*, *pack*, *sort*, and *scan*. Some flags for these commands can be specified in the *MH* profile; **xmh** may override them. The application resource **debug** can be set to true to see how **xmh** uses *MH* commands.

ENVIRONMENT

HOME - users's home directory
MH - to get the location of the *MH* profile file

FILES

~/*.mh_profile* - *MH* profile, used if the *MH* environment variable is not set
~/Mail - directory of folders, used if the *MH* profile cannot be found
~/*.xmhcheck* - optional, for multiple mail drops in cooperation with *slocal*.
/usr/local/*mh6* - *MH* commands, as a last resort, see **mhPath**.
~/Mail/<folder>/*.xmhcache* - *scan* output in each folder
~/Mail/<folder>/*.mh_sequences* - sequence definitions, in each folder
/tmp - temporary files, see **tempDir**.

SEE ALSO

X11(7), **xrdb(1)**, X Toolkit Intrinsic, Athena Widget Set
At least one book has been published about *MH* and **xmh**.

BUGS

- When the user closes a window, all windows which are transient for that window should also be closed by **xmh**.
- When **XmhUseAsComposition** and **XmhViewUseAsComposition** operate on messages in the **DraftsFolder**, **xmh** disallows editing of the composition if the same message is also being viewed in another window.
- Occasionally after committing changes, the table of contents will appear to be completely blank when there are actually messages present. When this happens, refreshing the display, or typing Control-L in the table of contents, will often cause the correct listing to appear. If this doesn't work, force a rescan of the folder.
- Should recognize and use the "unseen" message-sequence.
- Should determine by itself if the user hasn't used *MH* before, and offer to create the *.mh_profile*, instead of hanging on *inc*.
- A few commands are missing (rename folder, resend message).
- WM_DELETE_WINDOW protocol doesn't work right when requesting deletion of the first toc and view, while trying to keep other **xmh** windows around.
- Doesn't support annotations when replying to messages.
- Doesn't allow folders to be shared without write permission.

- Doesn't recognize private sequences.
- *MH* will report that the *.mh_sequences* file is poorly formatted if any sequence definition in a particular folder contains more than *BUFSIZ* characters. **xmh** tries to capture these messages and display them when they occur, but it cannot correct the problem.

COPYRIGHT

Copyright 1988, 1989, Digital Equipment Corporation.
Copyright 1989, 1991 Massachusetts Institute of Technology
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Terry Weissman, formerly of Digital Western Research Laboratory
Donna Converse, MIT X Consortium

| | | | | | |
|------------------------------|--|---------------|---------------------------------|---------------|--|
| NAME | xmkmf – simple interface to the imake utility for generating X11 Makefiles | | | | |
| SYNOPSIS | xmkmf [topdir [curdir]] | | | | |
| DESCRIPTION | <p>The xmkmf command runs the imake command with the appropriate options to generate an X11 Makefile from an Imakefile in the current directory. The X Window System uses imake extensively for both full builds of the source tree and for external software. Two variables, TOPDIR and CURDIR, can be set to make referencing files using relative path names easier. For example, the following imake command can be used to build the Makefile in the directory lib/X (relative to the top of the sources):</p> <pre style="margin-left: 40px;">% ../../../../config/imake -I../../../../config \ -DTOPDIR=../../.. -DCURDIR=./lib/X</pre> <p>The above imake command can be similarly accomplished with the following xmkmf command:</p> <pre style="margin-left: 40px;">% xmkmf ../../.. ./lib/X</pre> | | | | |
| OPTIONS | <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">topdir</td> <td>Relative path to top of sources</td> </tr> <tr> <td>curdir</td> <td>Current directory relative to top of sources</td> </tr> </table> | topdir | Relative path to top of sources | curdir | Current directory relative to top of sources |
| topdir | Relative path to top of sources | | | | |
| curdir | Current directory relative to top of sources | | | | |
| ENVIRONMENT VARIABLES | Same as imake . | | | | |
| FILES | Imakefile machine-independent Makefile descriptions | | | | |
| SEE ALSO | imake(1) , make(1S) | | | | |

| | |
|--------------------|---|
| NAME | xmodmap – utility for modifying keymaps in X |
| SYNOPSIS | xmodmap [-options ...] [<i>filename</i>] |
| DESCRIPTION | The xmodmap program is used to edit and display the keyboard modifier map and keymap table that are used by client applications to convert event keycodes into keysyms. It is usually run from the user's session startup script to configure the keyboard according to personal tastes. |
| OPTIONS | <p>The following options may be used with xmodmap:</p> <ul style="list-style-type: none">-display <i>display</i> This option specifies the host and display to use.-help This option indicates that a brief description of the command line arguments should be printed on the standard error channel. This will be done whenever an unhandled argument is given to xmodmap.-grammar This option indicates that a help message describing the expression grammar used in files and with -e expressions should be printed on the standard error.-verbose This option indicates that xmodmap should print logging information as it parses its input.-quiet This option turns off the verbose logging. This is the default.-n This option indicates that xmodmap should not change the mappings, but should display what it would do, like make(1S) does when given this option.-e <i>expression</i> This option specifies an expression to be executed. Any number of expressions may be specified from the command line.-pm This option indicates that the current modifier map should be printed on the standard output.-pk This option indicates that the current keymap table should be printed on the standard output.-pke This option indicates that the current keymap table should be printed on the standard output in the form of expressions that can be fed back to xmodmap.-pp This option indicates that the current pointer map should be printed on the standard output.- A lone dash means that the standard input should be used as the input file. <p>The <i>filename</i> specifies a file containing xmodmap expressions to be executed. This file is usually kept in the user's home directory with a name like .xmodmaprc.</p> |

**EXPRESSION
GRAMMAR**

The **xmodmap** program reads a list of expressions and parses them all before attempting to execute any of them. This makes it possible to refer to keysyms that are being redefined in a natural way without having to worry as much about name conflicts.

keycode *NUMBER = KEYSYMNAME ...*

The list of keysyms is assigned to the indicated keycode (which may be specified in decimal, hex or octal and can be determined by running the *xev* program in the examples directory).

keysym *KEYSYMNAME = KEYSYMNAME ...*

The *KEYSYMNAME* on the left hand side is translated into matching keycodes used to perform the corresponding set of **keycode** expressions. The list of keysym names may be found in the header file *<X11/keysymdef.h>* (without the **XK_** prefix) or the keysym database */usr/openwin/lib/XKeysymDB*. Note that if the same keysym is bound to multiple keys, the expression is executed for each matching keycode.

clear *MODIFIERNAME*

This removes all entries in the modifier map for the given modifier, where valid name are: **Shift**, **Lock**, **Control**, **Mod1**, **Mod2**, **Mod3**, **Mod4**, and **Mod5** (case does not matter in modifier names, although it does matter for all other names). For example, "clear Lock" will remove all any keys that were bound to the shift lock modifier.

add *MODIFIERNAME = KEYSYMNAME ...*

This adds all keys containing the given keysyms to the indicated modifier map. The keysym names are evaluated after all input expressions are read to make it easy to write expressions to swap keys (see the **EXAMPLES** section).

remove *MODIFIERNAME = KEYSYMNAME ...*

This removes all keys containing the given keysyms from the indicated modifier map. Unlike **add**, the keysym names are evaluated as the line is read in. This allows you to remove keys from a modifier without having to worry about whether or not they have been reassigned.

pointer = default

This sets the pointer map back to its default settings (button 1 generates a code of 1, button 2 generates a 2, etc.).

pointer = NUMBER ...

This sets to pointer map to contain the indicated button codes. The list always starts with the first physical button.

Lines that begin with an exclamation point (!) are taken as comments.

If you want to change the binding of a modifier key, you must also remove it from the appropriate modifier map.

EXAMPLES

Many pointers are designed such that the first button is pressed using the index finger of the right hand. People who are left-handed frequently find that it is more comfortable to reverse the button codes that get generated so that the primary button is pressed using

the index finger of the left hand. This could be done on a 3 button pointer as follows:

```
% xmodmap -e "pointer = 3 2 1"
```

Many editor applications support the notion of Meta keys (similar to Control keys except that Meta is held down instead of Control). However, some servers do not have a Meta keysym in the default keymap table, so one needs to be added by hand. The following command will attach Meta to the Multi-language key (sometimes labeled Compose Character). It also takes advantage of the fact that applications that need a Meta key simply need to get the keycode and don't require the keysym to be in the first column of the keymap table. This means that applications that are looking for a Multi_key (including the default modifier map) won't notice any change.

```
% xmodmap -e "keysym Multi_key = Multi_key Meta_L"
```

One of the more simple, yet convenient, uses of **xmodmap** is to set the keyboard's "rubout" key to generate an alternate keysym. This frequently involves exchanging Backspace with Delete to be more comfortable to the user. If the *ttyModes* resource in **xterm(1)** is set as well, all terminal emulator windows will use the same key for erasing characters:

```
% xmodmap -e "keysym BackSpace = Delete"
% echo "XTerm*ttyModes: erase ^?" | xrdp -merge
```

Some keyboards do not automatically generate less than and greater than characters when the comma and period keys are shifted. This can be remedied with **xmodmap** by resetting the bindings for the comma and period with the following scripts:

```
!
! make shift-, be < and shift-. be >
!
keysym comma = comma less
keysym period = period greater
```

One of the more irritating differences between keyboards is the location of the Control and Shift Lock keys. A common use of **xmodmap** is to swap these two keys as follows:

```
!
! Swap Caps_Lock and Control_L
!
remove Lock = Caps_Lock
remove Control = Control_L
keysym Control_L = Caps_Lock
keysym Caps_Lock = Control_L
add Lock = Caps_Lock
add Control = Control_L
```

The **keycode** command is useful for assigning the same keysym to multiple keycodes. Although unportable, it also makes it possible to write scripts that can reset the keyboard to a known state. The following script sets the backspace key to generate Delete (as

shown above), flushes all existing caps lock bindings, makes the CapsLock key be a control key, make F5 generate Escape, and makes Break/Reset be a shift lock.

! On the HP, the following keycodes have key caps as listed:

```
!
! 101 Backspace
! 55 Caps
! 14 Ctrl
! 15 Break/Reset
! 86 Stop
! 89 F5
!
```

```
keycode 101 = Delete
keycode 55 = Control_R
clear Lock
add Control = Control_R
keycode 89 = Escape
keycode 15 = Caps_Lock
add Lock = Caps_Lock
```

ENVIRONMENT

DISPLAY

to get default host and display number.

SEE ALSO

X11(7), Xlib documentation on key and pointer events

BUGS

Every time a **keycode** expression is evaluated, the server generates a **MappingNotify** event on every client. This can cause some thrashing. All of the changes should be batched together and done at once. Clients that receive keyboard input and ignore **MappingNotify** events will not notice any changes made to keyboard mappings.

Xmodmap should generate "add" and "remove" expressions automatically whenever a keycode that is already bound to a modifier is changed.

There should be a way to have the **remove** expression accept keycodes as well as keysyms for those times when you really mess up your mappings.

COPYRIGHT

Copyright 1988, 1989, 1990 Massachusetts Institute of Technology.
Copyright 1987 Sun Microsystems, Inc.
See **X11 (7)** for a full statement of rights and permissions.

AUTHOR

Jim Fulton, MIT X Consortium, rewritten from an earlier version by David Rosenthal of Sun Microsystems.

NAME xpr – print an X window dump

SYNOPSIS **xpr** [**-device** *devtype*] [**-scale** *scale*] [**-height** *inches*] [**-width** *inches*] [**-left** *inches*] [**-top** *inches*] [**-header** *string*] [**-trailer** *string*] [**-landscape**] [**-portrait**] [**-plane** *number*] [**-gray**] [**-rv**] [**-compact**] [**-output** *filename*] [**-append** *filename*] [**-noff**] [**-split** *n*] [**-psfig**] [**-density** *dpi*] [**-cutoff** *level*] [**-noposition**] [**-gamma** *correction*] [**-render** *algorithm*] [**-slide**] [*filename*]

DESCRIPTION **xpr** takes as input a window dump file produced by **xwd**(1) and formats it 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. If no file argument is given, the standard input is used. By default, **xpr** prints the largest possible representation of the window on the output page. Options allow the user to add headers and trailers, specify margins, adjust the scale and orientation, and append multiple window dumps to a single output file. Output is to standard output unless **-output** is specified.

Command Options

-device *devtype*

Specifies the device on which the file will be printed. Currently supported:

| | |
|---------------|---|
| la100 | Digital LA100 |
| ljet | HP LaserJet series and other monochrome PCL devices such as ThinkJet, QuietJet, RuggedWriter, HP2560 series, and HP2930 series printers |
| ln03 | Digital LN03 |
| pjet | HP PaintJet (color mode) |
| pjetxl | HP HP PaintJet XL Color Graphics Printer (color mode) |
| pp | IBM PP3812 |
| ps | PostScript printer |

The default is PostScript. **-device lw** (LaserWriter) is equivalent to **-device ps** and is provided only for backwards compatibility.

-scale *scale*

Affects the size of the window on the page. The PostScript, LN03, and HP printers are able to translate each bit in a window pixel map into a grid of a specified size. For example each bit might translate into a 3x3 grid. This would be specified by **-scale 3**. By default a window is printed with the largest scale that will fit onto the page for the specified orientation.

-height *inches*

Specifies the maximum height of the page.

-width *inches*

Specifies the maximum width of the page.

-left *inches*

Specifies the left margin in inches. Fractions are allowed. By default the window

- is centered in the page.
- top** *inches*
Specifies the top margin for the picture in inches. Fractions are allowed.
 - header** *string*
Specifies a header string to be printed above the window.
 - trailer** *string*
Specifies a trailer string to be printed below the window.
 - landscape**
Forces the window to be printed in landscape mode. By default a window is printed such that its longest side follows the long side of the paper.
 - plane** *number*
Specifies which bit plane to use in an image. The default is to use the entire image and map values into black and white based on color intensities.
 - gray** *2 / 3 / 4*
Uses a simple 2x2, 3x3, or 4x4 gray scale conversion on a color image, rather than mapping to strictly black and white. This doubles, triples, or quadruples the effective width and height of the image.
 - portrait**
Forces the window to be printed in portrait mode. By default a window is printed such that its longest side follows the long side of the paper.
 - rv** Forces the window to be printed in reverse video.
 - compact**
Uses simple run-length encoding for compact representation of windows with lots of white pixels.
 - output** *filename*
Specifies an output file name. If this option is not specified, standard output is used.
 - append** *filename*
Specifies a filename previously produced by **xpr** to which the window is to be appended.
 - noff** When specified in conjunction with **-append**, the window will appear on the same page as the previous window.
 - split** *n* This option allows the user to split a window onto several pages. This might be necessary for very large windows that would otherwise cause the printer to overload and print the page in an obscure manner.
 - psfig** Suppress translation of the PostScript picture to the center of the page.
 - density** *dpi*
Indicates what dot-per-inch density should be used by the HP printer.
 - cutoff** *level*
Changes the intensity level where colors are mapped to either black or white for monochrome output on a LaserJet printer. The *level* is expressed as percentage of

full brightness. Fractions are allowed.

-noposition

This option causes header, trailer, and image positioning command generation to be bypassed for LaserJet, PaintJet and PaintJet XL printers.

-gamma *correction*

This changes the intensity of the colors printed by PaintJet XL printer. The *correction* is a floating point value in the range 0.00 to 3.00. Consult the operator's manual to determine the correct value for the specific printer.

-render *algorithm*

This allows PaintJet XL printer to render the image with the best quality versus performance tradeoff. Consult the operator's manual to determine which *algorithms* are available.

-slide This option allows overhead transparencies to be printed using the PaintJet and PaintJet XL printers.

SEE ALSO

xwd(1), xwud(1), X11(7)

LIMITATIONS

The current version of **xpr** can generally print out on the LN03 most X windows that are not larger than two-thirds of the screen. For example, it will be able to print out a large Emacs window, but it will usually fail when trying to print out the entire screen. The LN03 has memory limitations that can cause it to incorrectly print very large or complex windows. The two most common errors encountered are "band too complex" and "page memory exceeded." In the first case, a window may have a particular six pixel row that contains too many changes (from black to white to black). This will cause the printer to drop part of the line and possibly parts of the rest of the page. The printer will flash the number '1' on its front panel when this problem occurs. A possible solution to this problem is to increase the scale of the picture, or to split the picture onto two or more pages. The second problem, "page memory exceeded," will occur if the picture contains too much black, or if the picture contains complex half-tones such as the background color of a display. When this problem occurs the printer will automatically split the picture into two or more pages. It may flash the number '5' on its front panel. There is no easy solution to this problem. It will probably be necessary to either cut and paste, or to rework the application to produce a less complex picture.

There are several limitations on the LA100 support: the picture will always be printed in portrait mode, there is no scaling, and the aspect ratio will be slightly off.

Support for PostScript output currently cannot handle the **-append**, **-noff** or **-split** options.

The **-compact** option is *only* supported for PostScript output. It compresses white space but not black space, so it is not useful for reverse-video windows.

For color images, should map directly to PostScript image support.

HP PRINTERS

If no *-density* is specified on the command line 300 dots per inch will be assumed for *ljet* and 90 dots per inch for *pjet*. Allowable *density* values for a LaserJet printer are 300, 150, 100, and 75 dots per inch. Consult the operator's manual to determine densities supported by other printers.

If no *-scale* is specified the image will be expanded to fit the printable page area.

The default printable page area is 8x10.5 inches. Other paper sizes can be accommodated using the *-height* and *-width* options.

Note that a 1024x768 image fits the default printable area when processed at 100 dpi with *scale=1*, the same image can also be printed using 300 dpi with *scale=3* but will require considerably more data be transferred to the printer.

xpr may be tailored for use with monochrome PCL printers other than the LaserJet. To print on a ThinkJet (HP2225A) **xpr** could be invoked as:

```
xpr -density 96 -width 6.667 filename
```

or for black-and-white output to a PaintJet:

```
xpr -density 180 filename
```

The monochrome intensity of a pixel is computed as $0.30*R + 0.59*G + 0.11*B$. If a pixel's computed intensity is less than the *-cutoff* level it will print as white. This maps light-on-dark display images to black-on-white hardcopy. The default cutoff intensity is 50% of full brightness. Example: specifying *-cutoff 87.5* moves the white/black intensity point to 87.5% of full brightness.

A LaserJet printer must be configured with sufficient memory to handle the image. For a full page at 300 dots per inch approximately 2MB of printer memory is required.

Color images are produced on the PaintJet at 90 dots per inch. The PaintJet is limited to sixteen colors from its 330 color palette on each horizontal print line. **xpr** will issue a warning message if more than sixteen colors are encountered on a line. **xpr** will program the PaintJet for the first sixteen colors encountered on each line and use the nearest matching programmed value for other colors present on the line.

Specifying the *-rv*, reverse video, option for the PaintJet will cause black and white to be interchanged on the output image. No other colors are changed.

Multiplane images must be recorded by *xwd* in *ZPixmap* format. Single plane (monochrome) images may be in either *XYPixmap* or *ZPixmap* format.

Some PCL printers do not recognize image positioning commands. Output for these printers will not be centered on the page and header and trailer strings may not appear where expected.

The **-gamma** and **-render** options are supported only on the PaintJet XL printers.

The **-slide** option is not supported for LaserJet printers.

The **-split** option is not supported for HP printers.

The **-gray** option is not supported for HP or IBM printers.

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.

Copyright 1986, Marvin Solomon and the University of Wisconsin.

Copyright 1988, Hewlett Packard Company.

See **X11(7)** for a full statement of rights and permissions.

AUTHORS

Michael R. Gretzinger, MIT Project Athena, Jose Capo, MIT Project Athena (PP3812 support), Marvin Solomon, University of Wisconsin, Bob Scheifler, MIT, Angela Bock and E. Mike Durbin, Rich Inc. (grayscale), Larry Rupp, HP (HP printer support).

| | |
|-----------------|---|
| NAME | xprop – property displayer for X |
| SYNOPSIS | xprop [-help] [-grammar] [-id <i>id</i>] [-root] [-name <i>name</i>] [-frame] [-font <i>font</i>] [-display <i>display</i>] [-len <i>n</i>] [-notype] [-fs <i>file</i>] [-remove <i>property-name</i>] [-spy] [-f <i>atom format [dformat]]</i> * [<i>format [dformat] atom</i>]* |
| SUMMARY | The xprop utility is for displaying window and font properties in an X server. One window or font is selected using the command line arguments or possibly in the case of a window, by clicking on the desired window. A list of properties is then given, possibly with formatting information. |
| OPTIONS | <p>-help Print out a summary of command line options.</p> <p>-grammar Print out a detailed grammar for all command line options.</p> <p>-id <i>id</i> This argument allows the user to select window <i>id</i> on the command line rather than using the pointer to select the target window. This is very useful in debugging X applications where the target window is not mapped to the screen or where the use of the pointer might be impossible or interfere with the application.</p> <p>-name <i>name</i> This argument allows the user to specify that the window named <i>name</i> is the target window on the command line rather than using the pointer to select the target window.</p> <p>-font <i>font</i> This argument allows the user to specify that the properties of font <i>font</i> should be displayed.</p> <p>-root This argument specifies that X's root window is the target window. This is useful in situations where the root window is completely obscured.</p> <p>-display <i>display</i> This argument allows you to specify the server to connect to; see X(7).</p> <p>-len <i>n</i> Specifies that at most <i>n</i> bytes of any property should be read or displayed.</p> <p>-notype Specifies that the type of each property should not be displayed.</p> <p>-fs <i>file</i> Specifies that file <i>file</i> should be used as a source of more formats for properties.</p> <p>-frame Specifies that when selecting a window by hand (i.e. if none of -name, -root, or -id are given), look at the window manager frame (if any) instead of looking for the client window.</p> <p>-remove <i>property-name</i> Specifies the name of a property to be removed from the indicated window.</p> <p>-spy Examine window properties forever, looking for property change events.</p> |

-f *name format [dformat]*

Specifies that the *format* for *name* should be *format* and that the *dformat* for *name* should be *dformat*. If *dformat* is missing, "`= $0+\n`" is assumed.

DESCRIPTION

For each of these properties, its value on the selected window or font is printed using the supplied formatting information if any. If no formatting information is supplied, internal defaults are used. If a property is not defined on the selected window or font, "not defined" is printed as the value for that property. If no property list is given, all the properties possessed by the selected window or font are printed.

A window may be selected in one of four ways. First, if the desired window is the root window, the `-root` argument may be used. If the desired window is not the root window, it may be selected in two ways on the command line, either by id number such as might be obtained from *xwininfo*, or by name if the window possesses a name. The `-id` argument selects a window by id number in either decimal or hex (must start with 0x) while the `-name` argument selects a window by name.

The last way to select a window does not involve the command line at all. If none of `-font`, `-id`, `-name`, and `-root` are specified, a crosshairs cursor is displayed and the user is allowed to choose any visible window by pressing any pointer button in the desired window. If it is desired to display properties of a font as opposed to a window, the `-font` argument must be used.

Other than the above four arguments and the `-help` argument for obtaining help, and the `-grammar` argument for listing the full grammar for the command line, all the other command line arguments are used in specifying both the format of the properties to be displayed and how to display them. The `-len n` argument specifies that at most *n* bytes of any given property will be read and displayed. This is useful for example when displaying the cut buffer on the root window which could run to several pages if displayed in full.

Normally each property name is displayed by printing first the property name then its type (if it has one) in parentheses followed by its value. The `-notype` argument specifies that property types should not be displayed. The `-fs` argument is used to specify a file containing a list of formats for properties while the `-f` argument is used to specify the format for one property.

The formatting information for a property actually consists of two parts, a *format* and a *dformat*. The *format* specifies the actual formatting of the property (i.e., is it made up of words, bytes, or longs?, etc.) while the *dformat* specifies how the property should be displayed.

The following paragraphs describe how to construct *formats* and *dformats*. However, for the vast majority of users and uses, this should not be necessary as the built in defaults contain the *formats* and *dformats* necessary to display all the standard properties. It should only be necessary to specify *formats* and *dformats* if a new property is being dealt with or the user dislikes the standard display format. New users especially are encouraged to skip this part.

A *format* consists of one of 0, 8, 16, or 32 followed by a sequence of one or more format characters. The 0, 8, 16, or 32 specifies how many bits per field there are in the property. Zero is a special case meaning use the field size information associated with the property itself. (This is only needed for special cases like type INTEGER which is actually three different types depending on the size of the fields of the property)

A value of 8 means that the property is a sequence of bytes while a value of 16 would mean that the property is a sequence of words. The difference between these two lies in the fact that the sequence of words will be byte swapped while the sequence of bytes will not be when read by a machine of the opposite byte order of the machine that originally wrote the property. For more information on how properties are formatted and stored, consult the Xlib manual.

Once the size of the fields has been specified, it is necessary to specify the type of each field (i.e., is it an integer, a string, an atom, or what?) This is done using one format character per field. If there are more fields in the property than format characters supplied, the last character will be repeated as many times as necessary for the extra fields. The format characters and their meaning are as follows:

- a The field holds an atom number. A field of this type should be of size 32.
- b The field is a boolean. A 0 means false while anything else means true.
- c The field is an unsigned number, a cardinal.
- i The field is a signed integer.
- m The field is a set of bit flags, 1 meaning on.
- s This field and the next ones until either a 0 or the end of the property represent a sequence of bytes. This format character is only usable with a field size of 8 and is most often used to represent a string.
- x The field is a hex number (like 'c' but displayed in hex - most useful for displaying window ids and the like)

An example *format* is 32ica which is the format for a property of three fields of 32 bits each, the first holding a signed integer, the second an unsigned integer, and the third an atom.

The format of a *dformat* unlike that of a *format* is not so rigid. The only limitations on a *dformat* is that one may not start with a letter or a dash. This is so that it can be distinguished from a property name or an argument. A *dformat* is a text string containing special characters instructing that various fields be printed at various points in a manner similar to the formatting string used by printf. For example, the *dformat* " is (\$0, \$1 \)\n" would render the POINT 3, -4 which has a *format* of 32ii as " is (3, -4)\n".

Any character other than a \$, ?, \, or a (in a *dformat* prints as itself. To print out one of \$, ?, \, or (precede it by a \. For example, to print out a \$, use \\$. Several special backslash sequences are provided as shortcuts. \n will cause a newline to be displayed while \t will cause a tab to be displayed. \o where o is an octal number will display character number o.

A \$ followed by a number *n* causes field number *n* to be displayed. The format of the displayed field depends on the formatting character used to describe it in the corresponding *format*. I.e., if a cardinal is described by 'c' it will print in decimal while if it is described by a 'x' it is displayed in hex.

If the field is not present in the property (this is possible with some properties), <field not available> is displayed instead. $\$n+$ will display field number *n* then a comma then field number *n+1* then another comma then ... until the last field defined. If field *n* is not defined, nothing is displayed. This is useful for a property that is a list of values.

A ? is used to start a conditional expression, a kind of if-then statement. $?exp(text)$ will display *text* if and only if *exp* evaluates to non-zero. This is useful for two things. First, it allows fields to be displayed if and only if a flag is set. And second, it allows a value such as a state number to be displayed as a name rather than as just a number. The syntax of *exp* is as follows:

```
exp ::= term | term=exp | !exp
term ::= n | $n | mn
```

The ! operator is a logical "not", changing 0 to 1 and any non-zero value to 0. = is an equality operator. Note that internally all expressions are evaluated as 32 bit numbers so -1 is not equal to 65535. = returns 1 if the two values are equal and 0 if not. *n* represents the constant value *n* while $\$n$ represents the value of field number *n*. *mn* is 1 if flag number *n* in the first field having format character 'm' in the corresponding *format* is 1, 0 otherwise.

Examples: $?m3(count: \$3\backslash n)$ displays field 3 with a label of count if and only if flag number 3 (count starts at 0!) is on. $?\$2=0(True)?!\$2=0(False)$ displays the inverted value of field 2 as a boolean.

In order to display a property, *xprop* needs both a *format* and a *dformat*. Before *xprop* uses its default values of a *format* of 32x and a *dformat* of " = { \$0+ }\n", it searches several places in an attempt to find more specific formats. First, a search is made using the name of the property. If this fails, a search is made using the type of the property. This allows type STRING to be defined with one set of formats while allowing property WM_NAME which is of type STRING to be defined with a different format. In this way, the display formats for a given type can be overridden for specific properties.

The locations searched are in order: the format if any specified with the property name (as in 8x WM_NAME), the formats defined by -f options in last to first order, the contents of the file specified by the -fs option if any, the contents of the file specified by the environmental variable XPROPFORMATS if any, and finally *xprop*'s built in file of formats.

The format of the files referred to by the -fs argument and the XPROPFORMATS variable is one or more lines of the following form:

```
name format [dformat]
```

Where *name* is either the name of a property or the name of a type, *format* is the *format* to be used with *name* and *dformat* is the *dformat* to be used with *name*. If *dformat* is not present, " = \$0+\n" is assumed.

EXAMPLES

To display the name of the root window: *xprop -root WM_NAME*

To display the window manager hints for the clock: *xprop -name xclock WM_HINTS*

To display the start of the cut buffer: *xprop -root -len 100 CUT_BUFFER0*

To display the point size of the fixed font: *xprop -font fixed POINT_SIZE*

To display all the properties of window # 0x200007: *xprop -id 0x200007*

ENVIRONMENT**DISPLAY**

To get default display.

XPROPFORMATS

Specifies the name of a file from which additional formats are to be obtained.

SEE ALSO

X11(7), **xwininfo(1)**

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.

See **X(7)** for a full statement of rights and permissions.

AUTHOR

Mark Lillibridge, MIT Project Athena

| | |
|--------------------|--|
| NAME | xrdb – X server resource database utility |
| SYNOPSIS | xrdb [-option ...] [<i>filename</i>] |
| DESCRIPTION | <p>Xrdb is used to get or set the contents of the RESOURCE_MANAGER property on the root window of screen 0, or the SCREEN_RESOURCES property on the root window of any or all screens, or everything combined. You would normally run this program from your X startup file.</p> <p>Most X clients use the RESOURCE_MANAGER and SCREEN_RESOURCES properties to get user preferences about color, fonts, and so on for applications. Having this information in the server (where it is available to all clients) instead of on disk, solves the problem in previous versions of X that required you to maintain <i>defaults</i> files on every machine that you might use. It also allows for dynamic changing of defaults without editing files.</p> <p>The RESOURCE_MANAGER property is used for resources that apply to all screens of the display. The SCREEN_RESOURCES property on each screen specifies additional (or overriding) resources to be used for that screen. (When there is only one screen, SCREEN_RESOURCES is normally not used, all resources are just placed in the RESOURCE_MANAGER property.)</p> <p>The file specified by <i>filename</i> (or the contents from standard input if - or no filename is given) is optionally passed through the C preprocessor with the following symbols defined, based on the capabilities of the server being used:</p> <p>BITS_PER_RGB=num the number of significant bits in an RGB color specification. This is the log base 2 of the number of distinct shades of each primary that the hardware can generate. Note that it usually is not related to PLANES.</p> <p>CLASS=visualclass one of StaticGray, GrayScale, StaticColor, PseudoColor, TrueColor, DirectColor. This is the visual class of the root window of the default screen.</p> <p>COLOR defined only if CLASS is one of StaticColor, PseudoColor, TrueColor, or DirectColor.</p> <p>HEIGHT=num the height of the default screen in pixels.</p> <p>SERVERHOST=hostname the hostname portion of the display to which you are connected.</p> <p>HOST=hostname the same as SERVERHOST.</p> <p>CLIENTHOST=hostname the name of the host on which xrdb is running.</p> <p>PLANES=num</p> |

the number of bit planes (the depth) of the root window of the default screen.

RELEASE=num

the vendor release number for the server. The interpretation of this number will vary depending on VENDOR.

REVISION=num

the X protocol minor version supported by this server (currently 0).

VERSION=num

the X protocol major version supported by this server (should always be 11).

VENDOR=vendor

a string specifying the vendor of the server.

WIDTH=num

the width of the default screen in pixels.

X_RESOLUTION=num

the x resolution of the default screen in pixels per meter.

Y_RESOLUTION=num

the y resolution of the default screen in pixels per meter.

Lines that begin with an exclamation mark (!) are ignored and may be used as comments.

Note that since **xrdp** can read from standard input, it can be used to change the contents of properties directly from a terminal or from a shell script.

OPTIONS

xrdp program accepts the following options:

-help This option (or any unsupported option) will cause a brief description of the allowable options and parameters to be printed.

-display *display*

This option specifies the X server to be used; see X(7). It also specifies the screen to use for the *-screen* option, and it specifies the screen from which preprocessor symbols are derived for the *-global* option.

-all This option indicates that operation should be performed on the screen-independent resource property (RESOURCE_MANAGER), as well as the screen-specific property (SCREEN_RESOURCES) on every screen of the display. For example, when used in conjunction with *-query*, the contents of all properties are output. For *-load* and *-merge*, the input file is processed once for each screen. The resources which occur in common in the output for every screen are collected, and these are applied as the screen-independent resources. The remaining resources are applied for each individual per-screen property. This the default mode of operation.

-global This option indicates that the operation should only be performed on the screen-independent RESOURCE_MANAGER property.

-screen This option indicates that the operation should only be performed on the SCREEN_RESOURCES property of the default screen of the display.

-screens

- This option indicates that the operation should be performed on the `SCREEN_RESOURCES` property of each screen of the display. For `-load` and `-merge`, the input file is processed for each screen.
- n** This option indicates that changes to the specified properties (when used with `-load` or `-merge`) or to the resource file (when used with `-edit`) should be shown on the standard output, but should not be performed.
 - quiet** This option indicates that warning about duplicate entries should not be displayed.
 - cpp *filename***
This option specifies the pathname of the C preprocessor program to be used. Although **xrdp** was designed to use CPP, any program that acts as a filter and accepts the `-D`, `-I`, and `-U` options may be used.
 - nocpp** This option indicates that **xrdp** should not run the input file through a preprocessor before loading it into properties.
 - symbols**
This option indicates that the symbols that are defined for the preprocessor should be printed onto the standard output.
 - query** This option indicates that the current contents of the specified properties should be printed onto the standard output. Note that since preprocessor commands in the input resource file are part of the input file, not part of the property, they won't appear in the output from this option. The **-edit** option can be used to merge the contents of properties back into the input resource file without damaging preprocessor commands.
 - load** This option indicates that the input should be loaded as the new value of the specified properties, replacing whatever was there (i.e. the old contents are removed). This is the default action.
 - merge** This option indicates that the input should be merged with, instead of replacing, the current contents of the specified properties. Note that this option does a lexicographic sorted merge of the two inputs, which is almost certainly not what you want, but remains for backward compatibility.
 - remove**
This option indicates that the specified properties should be removed from the server.
 - retain** This option indicates that the server should be instructed not to reset if **xrdp** is the first client. This never be necessary under normal conditions, since **xdm(1)** and **xinit(1)** always act as the first client.
 - edit *filename***
This option indicates that the contents of the specified properties should be edited into the given file, replacing any values already listed there. This allows you to put changes that you have made to your defaults back into your resource file, preserving any comments or preprocessor lines.
 - backup *string***

This option specifies a suffix to be appended to the filename used with `-edit` to generate a backup file.

`-Dname[=value]`

This option is passed through to the preprocessor and is used to define symbols for use with conditionals such as `#ifdef`.

`-Uname` This option is passed through to the preprocessor and is used to remove any definitions of this symbol.

`-Idirectory`

This option is passed through to the preprocessor and is used to specify a directory to search for files that are referenced with `#include`.

FILES Generalizes `~/Xdefaults` files.

SEE ALSO `X11(7)`, *Xlib Resource Manager* documentation, Xt resource documentation

ENVIRONMENT **DISPLAY**
to figure out which display to use.

BUGS The default for no arguments should be to query, not to overwrite, so that it is consistent with other programs.

WARNINGS By default, `xrdp` passes its input through the `cpp(1)` pre-processor. The `cpp` pre-processor defines the symbol "sun" with the value "1". This means that any unquoted appearance of the string "sun" in the input will be converted to the string "1". To avoid this behavior, invoke the command with the option `-nocpp` or the option `-Usun`. Alternatively, precede a group of lines that contain the string "sun" with the line `#undef sun` and follow them by `#define sun 1`

COPYRIGHT Copyright 1991, Digital Equipment Corporation and MIT.

AUTHORS Bob Scheifler, Phil Karlton, rewritten from the original by Jim Gettys

| | |
|--------------------|--|
| NAME | xrefresh – refresh all or part of an X screen |
| SYNOPSIS | xrefresh [-option ...] |
| DESCRIPTION | Xrefresh is a simple X program that causes all or part of your screen to be repainted. This is useful when system messages have messed up your screen. Xrefresh maps a window on top of the desired area of the screen and then immediately unmaps it, causing refresh events to be sent to all applications. By default, a window with no background is used, causing all applications to repaint “smoothly.” However, the various options can be used to indicate that a solid background (of any color) or the root window background should be used instead. |
| ARGUMENTS | <p>-white Use a white background. The screen just appears to flash quickly, and then repaint.</p> <p>-black Use a black background (in effect, turning off all of the electron guns to the tube). This can be somewhat disorienting as everything goes black for a moment.</p> <p>-solid <i>color</i> Use a solid background of the specified color. Try green.</p> <p>-root Use the root window background.</p> <p>-none This is the default. All of the windows simply repaint.</p> <p>-geometry <i>WxH+X+Y</i> Specifies the portion of the screen to be repainted; see <i>X(1)</i>.</p> <p>-display <i>display</i> This argument allows you to specify the server and screen to refresh; see <i>X(7)</i>.</p> |
| X DEFAULTS | <p>The <i>xrefresh</i> program uses the routine <i>XGetDefault(3X)</i> to read defaults, so its resource names are all capitalized.</p> <p>Black, White, Solid, None, Root Determines what sort of window background to use.</p> <p>Geometry Determines the area to refresh. Not very useful.</p> |
| ENVIRONMENT | DISPLAY - To get default host and display number. |
| SEE ALSO | X11(7) |
| BUGS | It should have just one default type for the background. |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |

AUTHORS

Jim Gettys, Digital Equipment Corp., MIT Project Athena

| | |
|--------------------|--|
| NAME | xscope – X Window Protocol Viewer |
| SYNOPSIS | xscope [-i <i>input-port</i>] [-o <i>output-port</i>] [-h <i>host</i>] [-d <i>display</i>] [-q] [-v <i>print-level</i>] |
| DESCRIPTION | <p>Xscope sits in-between an X11 client and an X11 server and prints the contents of each request, reply, error, or event that is communicated between them. This information can be useful in debugging and performance tuning of X11 servers and clients.</p> <p>To operate, xscope must know the host, port, and display to use to connect to the X11 server. In addition, it must know the port on which it should listen for X11 clients. Two cases are common:</p> <ol style="list-style-type: none"> (1) The X11 server is on the same host as xscope. In this case, the input port for xscope should be selected as an X11 server on a different display, and the client DISPLAY argument adjusted to select xscope. For example, if the X11 server is on port 6000, display 1, then xscope can use port 6002 as its input port. The client can use display 1 for direct access to X11 or display 2 for access to xscope. (2) The X11 server is on a different host than xscope. In this case the same input and output ports can be used, and the host component of the DISPLAY is used to select xscope or X11. |
| OPTIONS | <p>-i <i>input-port</i> Specify the port that xscope will use to take requests from clients (defaults to 1). For X11, this port is automatically biased by 6000.</p> <p>-o <i>output-port</i> Determines the port that xscope will use to connect to X11 (defaults to 0). For X11, this port is automatically biased by 6000.</p> <p>-h <i>host</i> Determines the host that xscope will use to find its X11 server.</p> <p>-d <i>display</i> Defines the display number. The display number is added to the input and output port to give the actual ports which are used by xscope.</p> <p>-q Quiet output mode. Gives only the names of requests, replies, errors, and events, but does not indicate contents.</p> <p>-v <i>print-level</i> Determines the level of printing which xscope will provide. The print-level can be 0 (same as quiet mode), 1, 2, 3, 4. The larger numbers give more and more output. For example, a successful setup returns a string which is the name of the vendor of the X11 server. At level 1, the explicit field giving the length of the string is suppressed since it can be inferred from the string. At level 2 and above the length is explicitly printed.</p> |
| EXAMPLES | <p>xscope -i1 -o0 /dev/null /tmp/xscope.out & client -display localhost:1</p> <p>This command would have xscope communicate with an X11 server on the local host, display 0; xscope itself would be available on the current host as display 1 (display of 0 plus the 1 of -i1). The standard input is redirected from <code>/dev/null</code> to prevent xscope</p> |

from stopping when put into the background. Output is redirected to a file in /tmp.

xscope -v4 -hcleo -d0 -o0 -i1

This command would have **xscope** communicate with an X11 server on host "cleo", display 0; **xscope** itself would be available on the current host as display 1 (display of 0 plus the 1 of -i1). Verbose level 4.

xscope -q -d1 -o1 -o3

The X11 server for the current host, display 2 (1 for -d1 plus 1 for -o1) would be used by **xscope** which would run as display 4 (1 for -d1 plus 3 for -o3). Quiet mode (verbose level 0).

SEE ALSO X11(7)

AUTHOR James L. Peterson (MCC)

| | |
|----------------------------|--|
| NAME | Xserver – X Window System server |
| SYNOPSIS | X [:displaynumber] [-option ...] [ttyname] |
| DESCRIPTION | X is the generic name for the X Window System server. It is frequently a link or a copy of the appropriate server binary for driving the most frequently used server on a given machine. |
| STARTING THE SERVER | <p>The server is usually started from the X Display Manager program xdm(1). This utility is run from the system boot files and takes care of keeping the server running, prompting for usernames and passwords, and starting up the user sessions. It is easily configured for sites that wish to provide nice, consistent interfaces for novice users (loading convenient sets of resources, starting up a window manager, clock, and nice selection of terminal emulator windows).</p> <p>Installations that run more than one window system will still need to use the xinit(1) utility. However, xinit is to be considered a tool for building startup scripts and is not intended for use by end users. Site administrators are strongly urged to use xdm, or build other interfaces for novice users.</p> <p>When the X server starts up, it takes over the display. If you are running on a workstation whose console is the display, you cannot log into the console while the server is running.</p> |
| NETWORK CONNECTIONS | <p>The X server supports connections made using the following reliable byte-streams:</p> <p><i>TCP/IP</i> Listen on port 6000+n, where <i>n</i> is the display number.</p> <p><i>Unix Domain</i> Use <i>/tmp/.X11-unix/Xn</i> as the filename for the socket, where <i>n</i> is the display number.</p> <p><i>DECnet</i> Respond to connections to object XSXn, where <i>n</i> is the display number. This is not supported in all environments.</p> |
| OPTIONS | <p>All X servers accept the following command line options:</p> <p>-a number Set pointer acceleration (i.e. the ratio of how much is reported to how much the user actually moved the pointer). The default is 2.</p> <p>-ac Disable host-based access control mechanisms. Enables access by any host, and permits any host to modify the access control list. Use with caution. This option exists primarily for executing remote test suites.</p> <p>-audit level Set the audit trail level. The default is 1; only connection rejections are reported. Level 2 also reports successful connections and disconnects. Level 0 turns off the audit trail. Audit output is sent on the standard error output stream.</p> |

- auth** *authorization-file*
Specify a file which contains a collection of authorization records used to authenticate access. See also **xdm(1)**
- bc** Disable certain kinds of error checking for bug compatibility with previous releases (e.g., to work around bugs in R2 and R3 xterms and toolkits). Deprecated.
- bs** Disable backing store support on all screens.
- c** Turn off key-click.
- c** *volume* Set key-click volume, range is 0-100.
- cc** *class*
Set the visual class for the root window of color screens. Class numbers are as specified in the X protocol. Not implemented in all servers.
- co** *filename*
Set name of RGB color database.
- core** The server dumps core on fatal errors.
- dpi** *resolution*
Set the resolution of the screen, in dots per inch. Use when the server cannot determine the screen size from the hardware. The default is 90.
- f** *volume*
Set beep (bell) volume, range is 0-100. The default is 50.
- fc** *cursorFont*
Set default cursor font.
- fn** *font* Set default font.
- fp** *fontPath*
Set the font search path. This path is a comma separated list of directories which the X server uses to search for font databases.
- ep** *encodingPath*
Set the font encoding search path.
- help** Print a usage message.
- I** All following command line arguments are ignored.
- logo** Turn on the X Window System logo display in the screen-saver. There is currently no way to change this from a client.
- nologo** Turn off the X Window System logo display in the screen-saver. There is currently no way to change this from a client.
- p** *minutes*
Set screen-saver pattern cycle time in minutes. The default is 10 minutes.
- pn** The server is to continue running if it fails to establish all of its well-known sockets, but establishes at least one.
- r** Turn off auto-repeat.

r Turn on auto-repeat.

-s *minutes*
Set screen-saver timeout time in minutes. The default is 10 minutes.

-su Disable save under support on all screens.

-t *number*
Set pointer acceleration threshold in pixels (i.e. after how many pixels pointer acceleration should take effect). The default is 4.

-terminate
The server is to terminate instead of resetting.

-to *seconds*
Set default connection timeout in seconds. The default is 60 seconds.

-tst Disable all testing extensions (e.g., XTEST, XTrap, XTestExtension1).

ttyxx ignored, for servers started the ancient way (from init).

v Set video-off screen-saver preference.

-v Set video-on screen-saver preference.

-wm Force the default backing-store of all windows to be WhenMapped; an inexpensive way of getting backing-store to apply to all windows.

-x *extension*
Load the specified extension at init. Not supported in most implementations.

You can also have the X server connect to **xdm(1)** using XDMCP. Although this is not typically useful as it does not allow **xdm** to manage the server process, it can be used to debug XDMCP implementations, and serves as a sample implementation of the server side of XDMCP. For more information on this protocol, see the *X Display Manager Control Protocol* specification. The following options control the behavior of XDMCP.

-query *host-name*
Enable XDMCP and send Query packets to the specified host.

-broadcast
Enable XDMCP and broadcast BroadcastQuery packets to the network. The first responding display manager will be chosen for the session.

-indirect *host-name*
Enable XDMCP and send IndirectQuery packets to the specified host.

-port *port-num*
Use an alternate port number for XDMCP packets. Must be specified before any **-query**, **-broadcast** or **-indirect** options.

-once Terminate the server after one session.

-class *display-class*
XDMCP has an additional display qualifier used in resource lookup for display-specific options. This option sets that value, by default it is "MIT-Unspecified" (not a very useful value).

-displayID *display-id*

Yet another XDMCP specific value, this one allows the display manager to identify each display so that it can locate the shared key.

Many servers also have device-specific command line options. See the manual pages for the individual servers for more details.

SECURITY

The X server implements a simple authorization protocol, MIT-MAGIC-COOKIE-1 which uses data private to authorized clients and the server. This is a rather trivial scheme; if the client passes authorization data which is the same as that held by the server, it is allowed connection access. This scheme is worse than the host-based access control mechanisms in environments with unsecure networks as it allows any host to connect, given that it has the private key. But in many environments, this level of security is better than the host-based scheme as it provides access control on a per-user instead of per-host basis.

In addition, the server provides support for a DES-based authorization scheme, SUN-DES-1, using Sun's Secure RPC. It involves encrypting data with the X server's public key. See the *OpenWindows Server Programmer's Guide* for more information.

The authorization data is passed to the server in a private file named with the `-auth` command line option. Before accepting connections after a reset or when the server is starting, it reads this file. If this file contains authorization records, the local host is not allowed access to the server; only clients which send one of the authorization records contained in the file in the connection setup information are allowed access. See the `xauth(1)` manual page for a description of the binary format of this file.

The X server also uses a host-based access control list for deciding whether or not to accept connections from clients on a particular machine. If no other authorization mechanism is being used, this list initially consists of the host on which the server is running as well as any machines listed in the file `/etc/Xn.hosts`, where `n` is the display number of the server. The file contains either an Internet hostname (e.g. `expo.lcs.mit.edu`) or a DECnet hostname in double colon format (e.g. `hydra::`). Each hostname must be newline separated with no leading or trailing whitespace. For example:

```
joesworkstation
corporate.company.com
star::
bigcpu::
```

Users add or remove hosts from this list and enable or disable access control using the `xhost` command from the same machine as the server.

The X protocol intrinsically does not have any notion of window operation permissions or place any restrictions on what a client can do; if a program can connect to a display, it has full run of the screen. Sites that have better authentication and authorization systems (such as Kerberos) might wish to make use of the hooks in the libraries and the server to provide additional security models.

SIGNALS

The X server handles the following signals:

SIGHUP

Close all existing connections, free all resources, and restore server defaults. This signal is sent by the display manager whenever the user's primary application (usually an **xterm**(1) or window manager) exits to force the server to clean up and prepare for the next user.

SIGTERM

The server exits cleanly.

SIGUSR1

This signal is used quite differently from either of the above. When the server starts, determines if SIGUSR1 is set to SIG_IGN. If this is true the server sends a SIGUSR1 to its parent process after it has set up the various connection schemes. **Xdm** uses this feature to recognize when the server is ready for client connections (see **xdm**(1)).

FONTS

Fonts are normally stored as individual files across various directories. The X server can obtain fonts from directories and/or from font servers. The list of directories and font servers the X server uses when trying to open a font is controlled by the *font path*.

Although most sites will choose to have the X server start up with the appropriate font path (see the **-fp**), This path can be overridden using the **xset**(1) program.

The default font path for the X server includes eight directories:

\$OPENWINHOME/lib/X11/fonts/misc

This directory contains many miscellaneous bitmap fonts that are useful on all systems. It contains a family of fixed-width fonts, a family of fixed-width fonts from Dale Schumacher, several Kana fonts from Sony Corporation, two JIS Kanji fonts, two Hangul fonts from Daewoo Electronics, two Hebrew fonts from Joseph Friedman, the standard cursor font, two cursor fonts from Digital Equipment Corporation, and cursor and glyph fonts from Sun Microsystems. It also has various font name aliases for the fonts, including **fixed** and **variable**.

\$OPENWINHOME/lib/X11/fonts/Speedo

This directory contains outline fonts for Bitstream's Speedo rasterizer. A single font face, contributed by Bitstream, Inc., in normal, bold, italic, and bold italic, is provided.

\$OPENWINHOME/lib/X11/fonts/75dpi

This directory contains bitmap fonts contributed by Adobe Systems, Inc., Digital Equipment Corporation, Bitstream, Inc., Bigelow and Holmes, and Sun Microsystems, Inc. for 75 dots per inch displays. An integrated selection of sizes, styles, and weights are provided for each family.

\$OPENWINHOME/lib/X11/fonts/100dpi

This directory contains 100 dots per inch versions of some of the fonts in the *75dpi* directory.

\$OPENWINHOME/lib/X11/fonts/F3

This directory contains scalable outline fonts for the F3 format. 57 typefaces are

present.

\$OPENWINHOME/lib/X11/fonts/F3bitmaps

This directory contains bitmaps in various point sized for the 57 F3 fonts.

\$OPENWINHOME/lib/X11/fonts/Xt+

This directory contains fonts used by OLIT.

\$OPENWINHOME/lib/X11/fonts/Type1

This directory contains outline Adobe Type1 fonts.

Font databases are created by executing the **mkfontdir**(1) program in the directory containing the compiled versions of the fonts (the **.pcf** files). Whenever fonts are added to a directory, **mkfontdir** should be executed so that the server can find the new fonts. If **mkfontdir** is not run, the server will not find any fonts in the directory.

DIAGNOSTICS

Too numerous to list them all. If run from **init**(1M), errors are typically logged in the file **/usr/adm/X*msgs**,

FILES

| | |
|--|-----------------------------|
| <i>/etc/X*.hosts</i> | Initial access control list |
| <i>\$OPENWINHOME/lib/X11/fonts/misc</i> | Bitmap font directory |
| <i>\$OPENWINHOME/lib/X11/fonts/75dpi</i> | Bitmap font directory |
| <i>\$OPENWINHOME/lib/X11/fonts/100dpi</i> | Bitmap font directory |
| <i>\$OPENWINHOME/lib/X11/fonts/Speedo</i> | Outline font directory |
| <i>\$OPENWINHOME/lib/X11/fonts/F3</i> | F3 outline font directory |
| <i>\$OPENWINHOME/lib/X11/fonts/F3bitmaps</i> | Bitmap font directory |
| <i>\$OPENWINHOME/lib/X11/fonts/Xt+</i> | OLIT font directory |
| <i>\$OPENWINHOME/lib/X11/fonts/Type1</i> | Outline font directories |
| <i>\$OPENWINHOME/lib/X11/fonts/PEX</i> | PEX font directories |
| <i>\$OPENWINHOME/lib/X11/rgb.txt</i> | Color database |
| <i>/tmp/.X11-unix/X*</i> | Unix domain socket |
| <i>/usr/adm/X*msgs</i> | Error log file |

| | |
|------------------|---|
| SEE ALSO | X11(7) , bdfpcf(1) , mkfontdir(1) , xauth(1) , xdm(1) , xhost(1) , xinit(1) , xset(1) , xsetroot(1) , xterm(1) , Xsun(1) <i>X Window System Protocol, Definition of the Porting Layer for the X v11 Sample Server, Strategies for Porting the X v11 Sample Server, Godzilla's Guide to Porting the X V11 Sample Server</i> |
| BUGS | The option syntax is inconsistent with itself and xset(1) . The acceleration option should take a numerator and a denominator like the protocol. If X dies before its clients, new clients won't be able to connect until all existing connections have their TCP TIME_WAIT timers expire. The color database is missing a large number of colors. |
| COPYRIGHT | Copyright 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991 Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHORS | The sample server was originally written by Susan Angebrannt, Raymond Drewry, Philip Karlton, and Todd Newman, from Digital Equipment Corporation, with support from a large cast. It has since been extensively rewritten by Keith Packard and Bob Scheifler, from MIT. |

| | |
|--------------------|--|
| NAME | xset – user preference utility for X |
| SYNOPSIS | xset [-display <i>display</i>] [-b] [b on/off] [b [<i>volume</i> [<i>pitch</i> [<i>duration</i>]]] [[-]bc] [-c] [c on/off] [c [<i>volume</i>]] [[-+]fp[-+=] <i>path</i> , <i>path</i> ,...] [fp default] [fp rehash] [[-]led [<i>integer</i>]] [led on/off] [m[ouse] [<i>accel_mult</i> / <i>accel_div</i>] [<i>threshold</i>]]] [m[ouse] default] [p <i>pixel color</i>] [[-]r] [r on/off] [s [<i>length</i> [<i>period</i>]]] [s blank/noblink] [s expose/noexpose] [s on/off] [s default] [q] |
| DESCRIPTION | This program is used to set various user preference options of the display. |
| OPTIONS | <p>-display <i>display</i> This option specifies the server to use; see <i>X(1)</i>.</p> <p>b The b option controls bell volume, pitch and duration. This option accepts up to three numerical parameters, a preceding dash(-), or a 'on/off' flag. If no parameters are given, or the 'on' flag is used, the system defaults will be used. If the dash or 'off' are given, the bell will be turned off. If only one numerical parameter is given, the bell volume will be set to that value, as a percentage of its maximum. Likewise, the second numerical parameter specifies the bell pitch, in hertz, and the third numerical parameter specifies the duration in milliseconds. Note that not all hardware can vary the bell characteristics. The X server will set the characteristics of the bell as closely as it can to the user's specifications.</p> <p>bc The <i>bc</i> option controls <i>bug compatibility</i> mode in the server, if possible; a preceding dash(-) disables the mode, otherwise the mode is enabled. Various pre-R4 clients pass illegal values in some protocol requests, and pre-R4 servers did not correctly generate errors in these cases. Such clients, when run against an R4 server, will terminate abnormally or otherwise fail to operate correctly. Bug compatibility mode explicitly reintroduces certain bugs into the X server, so that many such clients can still be run. This mode should be used with care; new application development should be done with this mode disabled. The server must support the MIT-SUNDRY-NONSTANDARD protocol extension in order for this option to work.</p> <p>c The c option controls key click. This option can take an optional value, a preceding dash(-), or an 'on/off' flag. If no parameter or the 'on' flag is given, the system defaults will be used. If the dash or 'off' flag is used, keyclick will be disabled. If a value from 0 to 100 is given, it is used to indicate volume, as a percentage of the maximum. The X server will set the volume to the nearest value that the hardware can support.</p> <p>fp= <i>path</i>,... The fp= sets the font path to the directories given in the <i>path</i> argument. The directories are interpreted by the server, not by the client, and are server-dependent. Directories that do not contain font databases created by <i>mkfontdir</i> will be ignored by the server.</p> <p>fp default</p> |

The **default** argument causes the font path to be reset to the server's default.

fp rehash

The **rehash** argument causes the server to reread the font databases in the current font path. This is generally only used when adding new fonts to a font directory (after running *mkfontdir* to recreate the font database).

-fp or fp-

The **-fp** and **fp-** options remove elements from the current font path. They must be followed by a comma-separated list of directories.

+fp or fp+

This **+fp** and **fp+** options prepend and append elements to the current font path, respectively. They must be followed by a comma-separated list of directories.

led

The **led** option controls the keyboard LEDs. This controls the turning on or off of one or all of the LEDs. It accepts an optional integer, a preceding dash(-) or an 'on/off' flag. If no parameter or the 'on' flag is given, all LEDs are turned on. If a preceding dash or the flag 'off' is given, all LEDs are turned off. If a value between 1 and 32 is given, that LED will be turned on or off depending on the existence of a preceding dash. A common LED which can be controlled is the "Caps Lock" LED. "xset led 3" would turn led #3 on. "xset -led 3" would turn it off. The particular LED values may refer to different LEDs on different hardware.

m

The **m** option controls the mouse parameters. The parameters for the mouse are 'acceleration' and 'threshold'. The acceleration can be specified as an integer, or as a simple fraction. The mouse, or whatever pointer the machine is connected to, will go 'acceleration' times as fast when it travels more than 'threshold' pixels in a short time. This way, the mouse can be used for precise alignment when it is moved slowly, yet it can be set to travel across the screen in a flick of the wrist when desired. One or both parameters for the **m** option can be omitted, but if only one is given, it will be interpreted as the acceleration. If no parameters or the flag 'default' is used, the system defaults will be set.

p

The **p** option controls pixel color values. The parameters are the color map entry number in decimal, and a color specification. The root background colors may be changed on some servers by altering the entries for BlackPixel and WhitePixel. Although these are often 0 and 1, they need not be. Also, a server may choose to allocate those colors privately, in which case an error will be generated. The map entry must not be a read-only color, or an error will result.

r

The **r** option controls the autorepeat. If a preceding dash or the 'off' flag is used, autorepeat will be disabled. If no parameters or the 'on' flag is used, autorepeat will be enabled.

s

The **s** option lets you set the screen saver parameters. This option accepts up to two numerical parameters, a 'blank/noblock' flag, an 'expose/noexpose' flag, an 'on/off' flag, or the 'default' flag. If no parameters or the 'default' flag is used, the system will be set to its default screen saver characteristics. The 'on/off' flags simply turn the screen saver functions on or off. The 'blank' flag

sets the preference to blank the video (if the hardware can do so) rather than display a background pattern, while 'noblack' sets the preference to display a pattern rather than blank the video. The 'expose' flag sets the preference to allow window exposures (the server can freely discard window contents), while 'noexpose' sets the preference to disable screen saver unless the server can regenerate the screens without causing exposure events. The length and period parameters for the screen saver function determines how long the server must be inactive for screen saving to activate, and the period to change the background pattern to avoid burn in. The arguments are specified in seconds. If only one numerical parameter is given, it will be used for the length.

q The **q** option gives you information on the current settings.

These settings will be reset to default values when you log out.

Note that not all X implementations are guaranteed to honor all of these options.

SEE ALSO X11(7) Xserver(1), xmodmap(1), xrdb(1), xsetroot(1)

COPYRIGHT Copyright 1988, Massachusetts Institute of Technology.
See X11(7) for a full statement of rights and permissions.

AUTHOR Bob Scheifler, MIT Laboratory for Computer Science
David Krikorian, MIT Project Athena (X11 version)

| | |
|--------------------|---|
| NAME | xsetroot – root window parameter setting utility for X |
| SYNOPSIS | xsetroot [-help] [-def] [-display <i>display</i>] [-cursor <i>cursorfile maskfile</i>] [-cursor_name <i>cursor-name</i>] [-bitmap <i>filename</i>] [-mod <i>x y</i>] [-gray] [-grey] [-fg <i>color</i>] [-bg <i>color</i>] [-rv] [-solid <i>color</i>] [-name <i>string</i>] |
| DESCRIPTION | <p>The setroot program allows you to tailor the appearance of the background ("root") window on a workstation display running X. Normally, you experiment with xsetroot until you find a personalized look that you like, then put the xsetroot command that produces it into your X startup file. If no options are specified, or if -def is specified, the window is reset to its default state. The -def option can be specified along with other options and only the non-specified characteristics will be reset to the default state.</p> <p>Only one of the background color/tiling changing options (-solid, -gray, -grey, -bitmap, and -mod) may be specified at a time.</p> |
| OPTIONS | <p>The various options are as follows:</p> <p>-help Print a usage message and exit.</p> <p>-def Reset unspecified attributes to the default values. (Restores the background to the familiar gray mesh and the cursor to the hollow x shape.)</p> <p>-cursor <i>cursorfile maskfile</i> This lets you change the pointer cursor to whatever you want when the pointer cursor is outside of any window. Cursor and mask files are bitmaps (little pictures), and can be made with the bitmap(1) program. You probably want the mask file to be all black until you get used to the way masks work.</p> <p>-cursor_name <i>cursorname</i> This lets you change the pointer cursor to one of the standard cursors from the cursor font. Refer to appendix B of the X protocol for the names (except that the XC_ prefix is elided for this option).</p> <p>-bitmap <i>filename</i> Use the bitmap specified in the file to set the window pattern. You can make your own bitmap files (little pictures) using the <i>bitmap(1)</i> program. The entire background will be made up of repeated "tiles" of the bitmap.</p> <p>-mod <i>x y</i> This is used if you want a plaid-like grid pattern on your screen. <i>x</i> and <i>y</i> are integers ranging from 1 to 16. Try the different combinations. Zero and negative numbers are taken as 1.</p> <p>-gray Make the entire background gray. (Easier on the eyes.)</p> <p>-grey Make the entire background grey.</p> <p>-fg <i>color</i> Use "color" as the foreground color. Foreground and background colors are meaningful only in combination with -cursor, -bitmap, or -mod.</p> <p>-bg <i>color</i></p> |

Use “color” as the background color.

-rv This exchanges the foreground and background colors. Normally the foreground color is black and the background color is white.

-solid *color*

This sets the background of the root window to the specified color. This option is only useful on color servers.

-name *string*

Set the name of the root window to “string”. There is no default value. Usually a name is assigned to a window so that the window manager can use a text representation when the window is iconified. This option is unused since you can’t iconify the background.

-display *display*

Specifies the server to connect to; see *X(1)*.

SEE ALSO

X11(7), **xset(1)**, **xrdb(1)**

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Mark Lillibridge, MIT Project Athena

| | |
|--------------------|---|
| NAME | xsol – play solitaire |
| SYNOPSIS | xsol [-display <i>connection</i>] [-nodrag] [[-,+] r] |
| DESCRIPTION | xsol plays a solitaire game simliar to Klondike. The playing field is made up of seven slots, where stacks are built in descending value with alternating suits. Aces are built on at the top, and ascending order in the same suit. Kings can be moved to any empty space in the playing field. The deck is gone through only once, card by card. The cards are moved using the Left mouse button (Button1). Pressing the button selects the card, and it (and any cards on it) can then by dragged to its destination, where releasing will place them. The deck cards are selected by clicking on them. |
| OPTIONS | -display <i>connection</i> Connect to X server display, <i>connection</i> . -nodrag A button press selects the card, and a second press places it. [-,+] r Turns reverse video on or off to make cards more readable on monochrome or gray scale devices. |
| SEE ALSO | X11(7) |
| COPYRIGHT | Copyright (c) 1988 by Sun Microsystems, Inc. David Lemke (lemke@sun.com) Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. |

| | |
|--------------------|--|
| NAME | xstdcmap – X standard colormap utility |
| SYNOPSIS | xstdcmap [-all] [-best] [-blue] [-default] [-delete <i>map</i>] [-display <i>display</i>] [-gray] [-green] [-help] [-red] [-verbose] |
| DESCRIPTION | The xstdcmap utility can be used to selectively define 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. Where at all possible, colormaps are created with read-only allocations. |
| OPTIONS | <p>The following options may be used with xstdcmap:</p> <p>-all This option indicates that all six standard colormap properties should be defined on each screen of the display. Not all screens will support visuals under which all six standard colormap properties are meaningful. xstdcmap will determine the best allocations and visuals for the colormap properties of a screen. Any previously existing standard colormap properties will be replaced.</p> <p>-best This option indicates that the RGB_BEST_MAP should be defined.</p> <p>-blue This option indicates that the RGB_BLUE_MAP should be defined.</p> <p>-default This option indicates that the RGB_DEFAULT_MAP should be defined.</p> <p>-delete <i>map</i> This option specifies that a standard colormap property should be removed. <i>map</i> may be one of: default, best, red, green, blue, or gray.</p> <p>-display <i>display</i> This option specifies the host and display to use; see X11(7).</p> <p>-gray This option indicates that the RGB_GRAY_MAP should be defined.</p> <p>-green This option indicates that the RGB_GREEN_MAP should be defined.</p> <p>-help This option indicates that a brief description of the command line arguments should be printed on the standard error. This will be done whenever an unhandled argument is given to <i>xstdcmap</i>.</p> <p>-red This option indicates that the RGB_RED_MAP should be defined.</p> <p>-verbose This option indicates that xstdcmap should print logging information as it parses its input and defines the standard colormap properties.</p> |
| ENVIRONMENT | DISPLAY to get default host and display number. |
| SEE ALSO | X11(7) |

COPYRIGHT

Copyright 1989, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Donna Converse, MIT X Consortium

| | |
|--------------------|---|
| NAME | Xsun – Solaris server for X Version 11 |
| SYNOPSIS | Xsun [option] ... |
| DESCRIPTION | Xsun is the Solaris server for Version 11 of the X window system on Solaris hardware. It is normally started by xinit(1) via openwin(1) . |
| OPTIONS | <p>In addition to the options described in Xserver(1), Xsun accepts the following command-line switches:</p> <p>-dev <i>filename</i></p> <p>This option specifies the name of the framebuffer device file to be used instead of the default framebuffer <code>/dev/fb</code>. Multiple instances of this option indicate multiple screens on the same server. After each -dev option a list of modifiers changes the behavior of the named device.</p> <p>[<i>left</i> <i>right</i> <i>top</i> <i>bottom</i>]</p> <p>Specify the position of a given screen in relation to the previous one on the command line. The default <i>right</i>.</p> <p><i>dpix</i> <i>n</i> The dpi in the x direction for this screen is <i>n</i>. The default is 90.</p> <p><i>dpiy</i> <i>n</i> The dpi in the y direction for this screen is <i>n</i>. The default is 90.</p> <p><i>defclass</i> [<i>GrayScale</i> <i>StaticGray</i> <i>PseudoColor</i> <i>StaticColor</i> <i>DirectColor</i> <i>TrueColor</i>] Use the specified visual as the default visual. The default is device dependent.</p> <p><i>defdepth</i> <i>n</i></p> <p>A visual of depth <i>n</i> is the default visual. The default is device dependent.</p> <p><i>grayvis</i> Only report <i>GrayScale</i> and/or <i>StaticGray</i> visuals.</p> <p>The following is an example of the -dev option that might be used on a system with a <i>cg6</i> and a <i>bw2</i>:</p> <pre style="margin-left: 40px;">-dev /dev/cgsix0 defclass GrayScale -dev /dev/bwtwo0 right</pre> <p>The Xsun server also supports the format used by the X11R5 sample X Server. Multiple screen systems are specified by using the following syntax on the command line:</p> <pre style="margin-left: 40px;">-dev <device 1>:<device 2>:...:<device n></pre> <p>The server uses device 1 as screen 0, device 2 as screen 1, etc., and the server assumes that screens are ordered left to right in ascending screen number. This means that the cursor moves off the right side of screen <i>n</i> and onto the left side of screen <i>n</i> + 1. When this format is used, no other -dev options are valid.</p> |

- accessX**
This option enables activation of the slowkey and stickykey functionality of the *AccessX* extension using the shift key on the keyboard.
- ar1 milliseconds**
Specify amount of time in milliseconds before a pressed key begins to autorepeating. The default is 500 milliseconds.
- ar2 milliseconds**
specify the interval in milliseconds between autorepeats of pressed keys. The default is 50 milliseconds.
- banner**
Display the OpenWindows banner screen at startup. The banner is displayed by default.
- dur milliseconds**
Set the duration of the bell in units of milliseconds. Default is 100.
- dpsfileops**
Allow the Display PostScript file operators access to the UNIX file system.
- flipPixels**
Reverse black and white pixel locations in the colormap. This is not the same as a reverse video option.
- pit percentage**
Set the percentage of the maximum pitch available to the hardware. Sun hardware does not support alternative pitch values.
- mden denominator**
Set the pointer acceleration denominator. The acceleration numerator is set with the *-a* option described in **Xserver(1)**. This permits fractional acceleration such as 3/2 or 1.5. Default value is 1.
- nobanner**
Do not display the OpenWindows banner screen at startup.
- nominexp**
This option is used to disable "minimized exposure", which is used only by multi-planegroup devices such as *gt*, *cg12* and *cg8*. "Minimized Exposure" means that the server will not send *expose* events to windows in one plane-group that are exposed by windows in another plane-group. By default the minimized exposure feature is turned ON.
- sharedretainedpath *directory_path***
This option is currently supported only for Sun internal software APIs.

SEE ALSO**openwin(1), X11(7), Xserver(1), xdm(1), xinit(1)**

| | |
|-----------------------|--|
| NAME | xterm – terminal emulator for X |
| SYNOPSIS | xterm [<i>-toolkitoption ...</i>] [<i>-option ...</i>] |
| DESCRIPTION | <p>The xterm program is a terminal emulator for the X Window System. It provides DEC VT102 and Tektronix 4014 compatible terminals for programs that can't use the window system directly. If the underlying operating system supports terminal resizing capabilities (for example, the SIGWINCH signal in systems derived from 4.3bsd), xterm will use the facilities to notify programs running in the window whenever it is resized.</p> <p>The VT102 and Tektronix 4014 terminals each have their own window so that you can edit text in one and look at graphics in the other at the same time. To maintain the correct aspect ratio (height/width), Tektronix graphics will be restricted to the largest box with a 4014's aspect ratio that will fit in the window. This box is located in the upper left area of the window.</p> <p>Although both windows may be displayed at the same time, one of them is considered the "active" window for receiving keyboard input and terminal output. This is the window that contains the text cursor. The active window can be chosen through escape sequences, the "VT Options" menu in the VT102 window, and the "Tek Options" menu in the 4014 window.</p> |
| EMULATIONS | <p>The VT102 emulation is fairly complete, but does not support the blinking character attribute nor the double-wide and double-size character sets. Terminfo entries that work with xterm include "xterm," "vt102," "vt100" and "ansi," and xterm automatically searches the /usr/share/lib/terminfo directory in this order for these entries and then sets the "TERM" environment variable (see terminfo(4)).</p> <p>Many of the special xterm features (like logging) may be modified under program control through a set of escape sequences different from the standard VT102 escape sequences. (See the "Xterm Control Sequences" document.)</p> <p>The Tektronix 4014 emulation is also fairly good. Four different font sizes and five different lines types are supported. The Tektronix text and graphics commands are recorded internally by xterm and may be written to a file by sending the COPY escape sequence (or through the Tektronix menu; see below). The name of the file will be "COPY<i>yy-MM-dd.hh:mm:ss</i>", where <i>yy</i>, <i>MM</i>, <i>dd</i>, <i>hh</i>, <i>mm</i> and <i>ss</i> are the year, month, day, hour, minute and second when the COPY was performed (the file is created in the directory xterm is started in, or the home directory for a login xterm).</p> |
| OTHER FEATURES | <p>Xterm automatically highlights the text cursor when the pointer enters the window (selected) and unhighlights it when the pointer leaves the window (unselected). If the window is the focus window, then the text cursor is highlighted no matter where the pointer is.</p> <p>In VT102 mode, there are escape sequences to activate and deactivate an alternate screen buffer, which is the same size as the display area of the window. When activated, the current screen is saved and replaced with the alternate screen. Saving of lines scrolled off</p> |

the top of the window is disabled until the normal screen is restored. The **terminfo(4)** entry for **xterm** allows the visual editor **vi(1)** to switch to the alternate screen for editing and to restore the screen on exit.

In either VT102 or Tektronix mode, there are escape sequences to change the name of the windows and to specify a new log file name. See *Xterm Control Sequences* for details. Enabling the escape sequence to change the log file name is a compile-time option; by default this escape sequence is ignored for security reasons.

OPTIONS

The **xterm** terminal emulator accepts all of the standard X Toolkit command line options as well as the following (if the option begins with a '+' instead of a '-', the option is restored to its default value):

- help** This causes **xterm** to print out a verbose message describing its options.
- 132** Normally, the VT102 DECCOLM escape sequence that switches between 80 and 132 column mode is ignored. This option causes the DECCOLM escape sequence to be recognized, and the **xterm** window will resize appropriately.
- ah** This option indicates that **xterm** should always highlight the text cursor. By default, **xterm** will display a hollow text cursor whenever the focus is lost or the pointer leaves the window.
- +ah** This option indicates that **xterm** should do text cursor highlighting based on focus.
- b number**
This option specifies the size of the inner border (the distance between the outer edge of the characters and the window border) in pixels. The default is 2.
- cc characterclassrange:value[,...]**
This sets classes indicated by the given ranges for using in selecting by words. See the section specifying character classes.
- cn** This option indicates that newlines should not be cut in line-mode selections.
- +cn** This option indicates that newlines should be cut in line-mode selections.
- cr color** This option specifies the color to use for text cursor. The default is to use the same foreground color that is used for text.
- cu** This option indicates that **xterm** should work around a bug in the **curses(3X)** cursor motion package that causes the **more(1)** program to display lines that are exactly the width of the window and are followed by a line beginning with a tab to be displayed incorrectly (the leading tabs are not displayed).
- +cu** This option indicates that that **xterm** should not work around the **curses(3X)** bug mentioned above.
- e program [arguments ...]**
This option specifies the program (and its command line arguments) to be run in the **xterm** window. It also sets the window title and icon name to be the basename of the program being executed if neither **-T** nor **-n** are given on the command line. **This must be the last option on the command line.**

- fb font** This option specifies a font to be used when displaying bold text. This font must be the same height and width as the normal font. If only one of the normal or bold fonts is specified, it will be used as the normal font and the bold font will be produced by overstriking this font. The default is to do overstriking of the normal font.
- j** This option indicates that **xterm** should do jump scrolling. Normally, text is scrolled one line at a time; this option allows **xterm** to move multiple lines at a time so that it doesn't fall as far behind. Its use is strongly recommended since it make **xterm** much faster when scanning through large amounts of text. The VT100 escape sequences for enabling and disabling smooth scroll as well as the "VT Options" menu can be used to turn this feature on or off.
- +j** This option indicates that **xterm** should not do jump scrolling.
- l** This option indicates that **xterm** should send all terminal output to a log file as well as to the screen. This option can be enabled or disabled using the "VT Options" menu.
- +l** This option indicates that **xterm** should not do logging.
- lf filename**
This option specifies the name of the file to which the output log described above is written. If *file* begins with a pipe symbol (|), the rest of the string is assumed to be a command to be used as the endpoint of a pipe. The ability to log to a pipe is a compile-time option which is disabled by default for security reasons. The default filename is "**XtermLog.XXXXXX**" (where XXXXXX is the process id of **xterm**) and is created in the directory from which **xterm** was started (or the user's home directory in the case of a login window).
- ls** This option indicates that the shell that is started in the **xterm** window be a login shell (i.e. the first character of argv[0] will be a dash, indicating to the shell that it should read the user's .login or .profile).
- +ls** This option indicates that the shell that is started should not be a login shell (i.e. it will be a normal "subshell").
- mb** This option indicates that **xterm** should ring a margin bell when the user types near the right end of a line. This option can be turned on and off from the "VT Options" menu.
- +mb** This option indicates that margin bell should not be rung.
- mc milliseconds**
This option specifies the maximum time between multi-click selections.
- ms color**
This option specifies the color to be used for the pointer cursor. The default is to use the foreground color.
- nb number**
This option specifies the number of characters from the right end of a line at which the margin bell, if enabled, will ring. The default is 10.

- rw** This option indicates that reverse-wraparound should be allowed. This allows the cursor to back up from the leftmost column of one line to the rightmost column of the previous line. This is very useful for editing long shell command lines and is encouraged. This option can be turned on and off from the “VT Options” menu.
- +rw** This option indicates that reverse-wraparound should not be allowed.
- aw** This option indicates that auto-wraparound should be allowed. This allows the cursor to automatically wrap to the beginning of the next line when when it is at the rightmost position of a line and text is output.
- +aw** This option indicates that auto-wraparound should not be allowed.
- s** This option indicates that **xterm** may scroll asynchronously, meaning that the screen does not have to be kept completely up to date while scrolling. This allows **xterm** to run faster when network latencies are very high and is typically useful when running across a very large internet or many gateways.
- +s** This option indicates that **xterm** should scroll synchronously.
- sb** This option indicates that some number of lines that are scrolled off the top of the window should be saved and that a scrollbar should be displayed so that those lines can be viewed. This option may be turned on and off from the “VT Options” menu.
- +sb** This option indicates that a scrollbar should not be displayed.
- sf** This option indicates that Sun Function Key escape codes should be generated for function keys.
- +sf** This option indicates that the standard escape codes should be generated for function keys.
- si** This option indicates that output to a window should not automatically reposition the screen to the bottom of the scrolling region. This option can be turned on and off from the “VT Options” menu.
- +si** This option indicates that output to a window should cause it to scroll to the bottom.
- sk** This option indicates that pressing a key while using the scrollbar to review previous lines of text should cause the window to be repositioned automatically in the normal position at the bottom of the scroll region.
- +sk** This option indicates that pressing a key while using the scrollbar should not cause the window to be repositioned.
- sl *number*** This option specifies the number of lines to save that have been scrolled off the top of the screen. The default is 64.
- t** This option indicates that **xterm** should start in Tektronix mode, rather than in VT102 mode. Switching between the two windows is done using the “Options” menus.
- +t** This option indicates that **xterm** should start in VT102 mode.

-tm string

This option specifies a series of terminal setting keywords followed by the characters that should be bound to those functions, similar to the *stty* program. Allowable keywords include: intr, quit, erase, kill, eof, eol, swtch, start, stop, brk, susp, dsusp, rprnt, flush, weras, and lnext. Control characters may be specified as $\hat{\text{char}}$ (e.g. $\hat{\text{c}}$ or $\hat{\text{u}}$) and $\hat{?}$ may be used to indicate delete.

-tn name

This option specifies the name of the terminal type to be set in the TERM environment variable. This terminal type must exist in the **termcap**(5) database and should have *li#* and *co#* entries.

-ut This option indicates that **xterm** shouldn't write a record into the the system log file */etc/utmp*.

+ut This option indicates that **xterm** should write a record into the system log file */etc/utmp*.

-vb This option indicates that a visual bell is preferred over an audible one. Instead of ringing the terminal bell whenever a Control-G is received, the window will be flashed.

+vb This option indicates that a visual bell should not be used.

-wf This option indicates that **xterm** should wait for the window to be mapped the first time before starting the subprocess so that the initial terminal size settings and environment variables are correct. It is the application's responsibility to catch subsequent terminal size changes.

+wf This option indicates that **xterm** show not wait before starting the subprocess.

-C This option indicates that this window should receive console output. This is not supported on all systems. To obtain console output, you must be the owner of the console device, and you must have read and write permission for it. If you are running X under *xdm* on the console screen you may need to have the session startup and reset programs explicitly change the ownership of the console device in order to get this option to work.

-Scn This option specifies the last two letters of the name of a pseudoterminal to use in slave mode, plus the number of the inherited file descriptor. The option is parsed " $\%c\%c\%d$ ". This allows **xterm** to be used as an input and output channel for an existing program and is sometimes used in specialized applications.

The following command line arguments are provided for compatibility with older versions. They may not be supported in the next release as the X Toolkit provides standard options that accomplish the same task.

%geom This option specifies the preferred size and position of the Tektronix window. It is shorthand for specifying the "**tekGeometry*" resource.

#geom This option specifies the preferred position of the icon window. It is shorthand for specifying the "**iconGeometry*" resource.

-T string

This option specifies the title for **xterm**. It is equivalent to **-title**.

-n *string*

This option specifies the icon name for **xterm**. It is shorthand for specifying the **iconName* resource. Note that this is not the same as the toolkit option **-name** (see below). The default icon name is the application name.

-r This option indicates that reverse video should be simulated by swapping the foreground and background colors. It is equivalent to **-reversevideo** or **-rv**.

-w *number*

This option specifies the width in pixels of the border surrounding the window. It is equivalent to **-borderwidth** or **-bw**.

The following standard X Toolkit command line arguments are commonly used with **xterm**:

-bg *color*

This option specifies the color to use for the background of the window. The default is "white."

-bd *color*

This option specifies the color to use for the border of the window. The default is "black."

-bw *number*

This option specifies the width in pixels of the border surrounding the window.

-fg *color*

This option specifies the color to use for displaying text. The default is "black."

-fn *font* This option specifies the font to be used for displaying normal text. The default is *fixed*.

-name *name*

This option specifies the application name under which resources are to be obtained, rather than the default executable file name. *Name* should not contain "." or "*" characters.

-title *string*

This option specifies the window title string, which may be displayed by window managers if the user so chooses. The default title is the command line specified after the **-e** option, if any, otherwise the application name.

-rv This option indicates that reverse video should be simulated by swapping the foreground and background colors.

-geometry *geometry*

This option specifies the preferred size and position of the VT102 window; see **X(7)**.

-display *display*

This option specifies the X server to contact; see **X(7)**.

-xrm *resourcestring*

This option specifies a resource string to be used. This is especially useful for

setting resources that do not have separate command line options.

-iconic This option indicates that **xterm** should ask the window manager to start it as an icon rather than as the normal window.

RESOURCES

The program understands all of the core X Toolkit resource names and classes as well as:

iconGeometry (class **IconGeometry**)

Specifies the preferred size and position of the application when iconified. It is not necessarily obeyed by all window managers.

termName (class **TermName**)

Specifies the terminal type name to be set in the TERM environment variable.

title (class **Title**)

Specifies a string that may be used by the window manager when displaying this application.

ttyModes (class **TtyModes**)

Specifies a string containing terminal setting keywords and the characters to which they may be bound. Allowable keywords include: intr, quit, erase, kill, eof, eol, swtch, start, stop, brk, susp, dsusp, rprnt, flush, weras, and lnext. Control characters may be specified as $\hat{\text{char}}$ (e.g. $\hat{\text{c}}$ or $\hat{\text{u}}$) and \^? may be used to indicate Delete. This is very useful for overriding the default terminal settings without having to do an *stty* every time an **xterm** is started.

utmpInhibit (class **UtmpInhibit**)

Specifies whether or not **xterm** should try to record the user's terminal in */etc/utmp*.

sunFunctionKeys (class **SunFunctionKeys**)

Specifies whether or not Sun Function Key escape codes should be generated for function keys instead of standard escape sequences.

waitForMap (class **WaitForMap**)

Specifies whether or not **xterm** should wait for the initial window map before starting the subprocess. The default is "false."

The following resources are specified as part of the *vt100* widget (class *VT100*):

allowSendEvents (class **AllowSendEvents**)

Specifies whether or not synthetic key and button events (generated using the X protocol *SendEvent* request) should be interpreted or discarded. The default is "false" meaning they are discarded. Note that allowing such events creates a very large security hole.

alwaysHighlight (class **AlwaysHighlight**)

Specifies whether or not **xterm** should always display a highlighted text cursor. By default, a hollow text cursor is displayed whenever the pointer moves out of the window or the window loses the input focus.

appcursorDefault (class **AppcursorDefault**)

If “true,” the cursor keys are initially in application mode. The default is “false.”

appkeypadDefault (class **AppkeypadDefault**)

If “true,” the keypad keys are initially in application mode. The default is “false.”

autoWrap (class **AutoWrap**)

Specifies whether or not auto-wraparound should be enabled. The default is “true.”

bellSuppressTime (class **BellSuppressTime**)

Number of milliseconds after a bell command is sent during which additional bells will be suppressed. Default is 200. If set non-zero, additional bells will also be suppressed until the server reports that processing of the first bell has been completed; this feature is most useful with the visible bell.

boldFont (class **BoldFont**)

Specifies the name of the bold font to use instead of overstriking.

c132 (class **C132**)

Specifies whether or not the VT102 DECCOLM escape sequence should be honored. The default is “false.”

charClass (class **CharClass**)

Specifies comma-separated lists of character class bindings of the form [*low-high:value*]. These are used in determining which sets of characters should be treated the same when doing cut and paste. See the section on specifying character classes.

curses (class **Curses**)

Specifies whether or not the last column bug in *curses*(3x) should be worked around. The default is “false.”

background (class **Background**)

Specifies the color to use for the background of the window. The default is “white.”

foreground (class **Foreground**)

Specifies the color to use for displaying text in the window. Setting the class name instead of the instance name is an easy way to have everything that would normally appear in the “text” color change color. The default is “black.”

cursorColor (class **Foreground**)

Specifies the color to use for the text cursor. The default is “black.”

eightBitInput (class **EightBitInput**)

If true, Meta characters input from the keyboard are presented as a single character with the eighth bit turned on. If false, Meta characters are converted into a two-character sequence with the character itself preceded by ESC. The default is “true.”

eightBitOutput (class **EightBitOutput**)

Specifies whether or not eight-bit characters sent from the host should be

accepted as is or stripped when printed. The default is “true.”

font (class **Font**)

Specifies the name of the normal font. The default is “fixed.”

font1 (class **Font1**)

Specifies the name of the first alternative font.

font2 (class **Font2**)

Specifies the name of the second alternative font.

font3 (class **Font3**)

Specifies the name of the third alternative font.

font4 (class **Font4**)

Specifies the name of the fourth alternative font.

font5 (class **Font5**)

Specifies the name of the fifth alternative font.

font6 (class **Font6**)

Specifies the name of the sixth alternative font.

geometry (class **Geometry**)

Specifies the preferred size and position of the VT102 window.

internalBorder (class **BorderWidth**)

Specifies the number of pixels between the characters and the window border.
The default is 2.

jumpScroll (class **JumpScroll**)

Specifies whether or not jump scroll should be used. The default is “true.”

logFile (class **Logfile**)

Specifies the name of the file to which a terminal session is logged. The default is “**XtermLog.XXXXX**” (where XXXXX is the process id of **xterm**).

logging (class **Logging**)

Specifies whether or not a terminal session should be logged. The default is “false.”

logInhibit (class **LogInhibit**)

Specifies whether or not terminal session logging should be inhibited. The default is “false.”

loginShell (class **LoginShell**)

Specifies whether or not the shell to be run in the window should be started as a login shell. The default is “false.”

marginBell (class **MarginBell**)

Specifies whether or not the bell should be run when the user types near the right margin. The default is “false.”

multiClickTime (class **MultiClickTime**)

Specifies the maximum time in milliseconds between multi-click select events.
The default is 250 milliseconds.

multiScroll (class **MultiScroll**)

Specifies whether or not scrolling should be done asynchronously. The default is "false."

nMarginBell (class **Column**)

Specifies the number of characters from the right margin at which the margin bell should be rung, when enabled.

pointerColor (class **Foreground**)

Specifies the foreground color of the pointer. The default is "XtDefaultForeground."

pointerColorBackground (class **Background**)

Specifies the background color of the pointer. The default is "XtDefaultBackground."

pointerShape (class **Cursor**)

Specifies the name of the shape of the pointer. The default is "xterm."

resizeGravity (class **ResizeGravity**)

Affects the behavior when the window is resized to be taller or shorter.

NorthWest specifies that the top line of text on the screen stay fixed. If the window is made shorter, lines are dropped from the bottom; if the window is made taller, blank lines are added at the bottom. This is compatible with the behavior in R4. **SouthWest** (the default) specifies that the bottom line of text on the screen stay fixed. If the window is made taller, additional saved lines will be scrolled down onto the screen; if the window is made shorter, lines will be scrolled off the top of the screen, and the top saved lines will be dropped.

reverseVideo (class **ReverseVideo**)

Specifies whether or not reverse video should be simulated. The default is "false."

reverseWrap (class **ReverseWrap**)

Specifies whether or not reverse-wraparound should be enabled. The default is "false."

saveLines (class **SaveLines**)

Specifies the number of lines to save beyond the top of the screen when a scrollbar is turned on. The default is 64.

scrollBar (class **ScrollBar**)

Specifies whether or not the scrollbar should be displayed. The default is "false."

scrollTtyOutput (class **ScrollCond**)

Specifies whether or not output to the terminal should automatically cause the scrollbar to go to the bottom of the scrolling region. The default is "true."

scrollKey (class **ScrollCond**)

Specifies whether or not pressing a key should automatically cause the scrollbar to go to the bottom of the scrolling region. The default is "false."

scrollLines (class **ScrollLines**)

Specifies the number of lines that the *scroll-back* and *scroll-forw* actions should use as a default. The default value is 1.

signalInhibit (class **SignalInhibit**)

Specifies whether or not the entries in the “Main Options” menu for sending signals to **xterm** should be disallowed. The default is “false.”

tekGeometry (class **Geometry**)

Specifies the preferred size and position of the Tektronix window.

tekInhibit (class **TekInhibit**)

Specifies whether or not Tektronix mode should be disallowed. The default is “false.”

tekSmall (class **TekSmall**)

Specifies whether or not the Tektronix mode window should start in its smallest size if no explicit geometry is given. This is useful when running **xterm** on displays with small screens. The default is “false.”

tekStartup (class **TekStartup**)

Specifies whether or not **xterm** should start up in Tektronix mode. The default is “false.”

titeInhibit (class **TiteInhibit**)

Specifies whether or not **xterm** should remove *ti* and *te* terminfo entries (used to switch between alternate screens on startup of many screen-oriented programs) from the TERM string. If set, **xterm** also ignores the escape sequence to switch to the alternate screen.

translations (class **Translations**)

Specifies the key and button bindings for menus, selections, “programmed strings,” etc. See **ACTIONS** below.

visualBell (class **VisualBell**)

Specifies whether or not a visible bell (i.e. flashing) should be used instead of an audible bell when Control-G is received. The default is “false.”

The following resources are specified as part of the *tek4014* widget (class *Tek4014*):

width (class **Width**)

Specifies the width of the Tektronix window in pixels.

height (class **Height**)

Specifies the height of the Tektronix window in pixels.

fontLarge (class **Font**)

Specifies the large font to use in the Tektronix window.

font2 (class **Font**)

Specifies font number 2 to use in the Tektronix window.

font3 (class **Font**)

Specifies font number 3 to use in the Tektronix window.

fontSmall (class **Font**)

Specifies the small font to use in the Tektronix window.

initialFont (class **InitialFont**)

Specifies which of the four Tektronix fonts to use initially. Values are the same as for the *set-tek-text* action. The default is "large."

ginTerminator (class **GinTerminator**)

Specifies what character(s) should follow a GIN report or status report. The possibilities are "none," which sends no terminating characters, "CRonly," which sends CR, and "CR&EOT," which sends both CR and EOT. The default is "none."

The resources that may be specified for the various menus are described in the documentation for the Athena **SimpleMenu** widget. The name and classes of the entries in each of the menus are listed below.

The *mainMenu* has the following entries:

securekbd (class **SmeBSB**)

This entry invokes the **secure()** action.

allowsends (class **SmeBSB**)

This entry invokes the **allow-send-events(toggle)** action.

logging (class **SmeBSB**)

This entry invokes the **set-logging(toggle)** action.

redraw (class **SmeBSB**)

This entry invokes the **redraw()** action.

line1 (class **SmeLine**)

This is a separator.

suspend (class **SmeBSB**)

This entry invokes the **send-signal(tstp)** action on systems that support job control.

continue (class **SmeBSB**)

This entry invokes the **send-signal(cont)** action on systems that support job control.

interrupt (class **SmeBSB**)

This entry invokes the **send-signal(int)** action.

hangup (class **SmeBSB**)

This entry invokes the **send-signal(hup)** action.

terminate (class **SmeBSB**)

This entry invokes the **send-signal(term)** action.

kill (class **SmeBSB**)

This entry invokes the **send-signal(kill)** action.

line2 (class **SmeLine**)

This is a separator.

quit (class **SmeBSB**)

This entry invokes the **quit()** action.

The *vtMenu* has the following entries:

scrollbar (class **SmeBSB**)

This entry invokes the **set-scrollbar(toggle)** action.

jumpscroll (class **SmeBSB**)

This entry invokes the **set-jumpscroll(toggle)** action.

reversevideo (class **SmeBSB**)

This entry invokes the **set-reverse-video(toggle)** action.

autowrap (class **SmeBSB**)

This entry invokes the **set-autowrap(toggle)** action.

reversewrap (class **SmeBSB**)

This entry invokes the **set-reversewrap(toggle)** action.

autolinefeed (class **SmeBSB**)

This entry invokes the **set-autolinefeed(toggle)** action.

appcursor (class **SmeBSB**)

This entry invokes the **set-appcursor(toggle)** action.

appkeypad (class **SmeBSB**)

This entry invokes the **set-appkeypad(toggle)** action.

scrollkey (class **SmeBSB**)

This entry invokes the **set-scroll-on-key(toggle)** action.

scrollttyoutput (class **SmeBSB**)

This entry invokes the **set-scroll-on-tty-output(toggle)** action.

allow132 (class **SmeBSB**)

This entry invokes the **set-allow132(toggle)** action.

cursesemul (class **SmeBSB**)

This entry invokes the **set-cursesemul(toggle)** action.

visualbell (class **SmeBSB**)

This entry invokes the **set-visualbell(toggle)** action.

marginbell (class **SmeBSB**)

This entry invokes the **set-marginbell(toggle)** action.

altscreen (class **SmeBSB**)

This entry is currently disabled.

line1 (class **SmeLine**)

This is a separator.

softreset (class **SmeBSB**)

This entry invokes the **soft-reset()** action.

hardreset (class **SmeBSB**)

This entry invokes the **hard-reset()** action.

clearsavedlines (class **SmeBSB**)

This entry invokes the **clear-saved-lines()** action.

line2 (class **SmeLine**)

This is a separator.

tekshow (class **SmeBSB**)

This entry invokes the **set-visibility(tek,toggle)** action.

tekmode (class **SmeBSB**)

This entry invokes the **set-terminal-type(tek)** action.

vhide (class **SmeBSB**)

This entry invokes the **set-visibility(vt,off)** action.

The *fontMenu* has the following entries:

fontdefault (class **SmeBSB**)

This entry invokes the **set-vt-font(d)** action.

font1 (class **SmeBSB**)

This entry invokes the **set-vt-font(1)** action.

font2 (class **SmeBSB**)

This entry invokes the **set-vt-font(2)** action.

font3 (class **SmeBSB**)

This entry invokes the **set-vt-font(3)** action.

font4 (class **SmeBSB**)

This entry invokes the **set-vt-font(4)** action.

font5 (class **SmeBSB**)

This entry invokes the **set-vt-font(5)** action.

font6 (class **SmeBSB**)

This entry invokes the **set-vt-font(6)** action.

fontescape (class **SmeBSB**)

This entry invokes the **set-vt-font(e)** action.

fontsel (class **SmeBSB**)

This entry invokes the **set-vt-font(s)** action.

The *tekMenu* has the following entries:

tektextlarge (class **SmeBSB**)

This entry invokes the **set-tek-text(l)** action.

tektext2 (class **SmeBSB**)

This entry invokes the **set-tek-text(2)** action.

tektext3 (class **SmeBSB**)

This entry invokes the **set-tek-text(3)** action.

tektextsmall (class **SmeBSB**)

This entry invokes the **set-tek-text(s)** action.

line1 (class **SmeLine**)

This is a separator.

tekpage (class **SmeBSB**)

This entry invokes the **tek-page()** action.

tekreset (class **SmeBSB**)

This entry invokes the **tek-reset()** action.

tekcopy (class **SmeBSB**)

This entry invokes the **tek-copy()** action.

line2 (class **SmeLine**)

This is a separator.

vtshow (class **SmeBSB**)

This entry invokes the **set-visibility(vt,toggle)** action.

vtmode (class **SmeBSB**)

This entry invokes the **set-terminal-type(vt)** action.

tekhide (class **SmeBSB**)

This entry invokes the **set-visibility(tek,toggle)** action.

The following resources are useful when specified for the Athena Scrollbar widget:

thickness (class **Thickness**)

Specifies the width in pixels of the scrollbar.

background (class **Background**)

Specifies the color to use for the background of the scrollbar.

foreground (class **Foreground**)

Specifies the color to use for the foreground of the scrollbar. The “thumb” of the scrollbar is a simple checkerboard pattern alternating pixels for foreground and background color.

POINTER USAGE

Once the VT102 window is created, **xterm** allows you to select text and copy it within the same or other windows.

The selection functions are invoked when the pointer buttons are used with no modifiers, and when they are used with the “shift” key. The assignment of the functions described below to keys and buttons may be changed through the resource database; see **ACTIONS** below.

Pointer button one (usually left) is used to save text into the cut buffer. Move the cursor to beginning of the text, and then hold the button down while moving the cursor to the end of the region and releasing the button. The selected text is highlighted and is saved in the global cut buffer and made the PRIMARY selection when the button is released. Double-clicking selects by words. Triple-clicking selects by lines. Quadruple-clicking goes back to characters, etc. Multiple-click is determined by the time from button up to button down, so you can change the selection unit in the middle of a selection. If the key/button bindings specify that an X selection is to be made, **xterm** will leave the selected text highlighted for as long as it is the selection owner.

Pointer button two (usually middle) ‘types’ (pastes) the text from the PRIMARY selection, if any, otherwise from the cut buffer, inserting it as keyboard input.

Pointer button three (usually right) extends the current selection. (Without loss of generality, you can swap “right” and “left” everywhere in the rest of this paragraph.) If pressed while closer to the right edge of the selection than the left, it extends/contracts the right edge of the selection. If you contract the selection past the left edge of the selection, **xterm** assumes you really meant the left edge, restores the original selection, then extends/contracts the left edge of the selection. Extension starts in the selection unit mode that the last selection or extension was performed in; you can multiple-click to cycle through them.

By cutting and pasting pieces of text without trailing new lines, you can take text from several places in different windows and form a command to the shell, for example, or take output from a program and insert it into your favorite editor. Since the cut buffer is globally shared among different applications, you should regard it as a ‘file’ whose contents you know. The terminal emulator and other text programs should be treating it as if it were a text file, i.e., the text is delimited by new lines.

The scroll region displays the position and amount of text currently showing in the window (highlighted) relative to the amount of text actually saved. As more text is saved (up to the maximum), the size of the highlighted area decreases.

Clicking button one with the pointer in the scroll region moves the adjacent line to the top of the display window.

Clicking button three moves the top line of the display window down to the pointer position.

Clicking button two moves the display to a position in the saved text that corresponds to the pointer’s position in the scrollbar.

Unlike the VT102 window, the Tektronix window does not allow the copying of text. It does allow Tektronix GIN mode, and in this mode the cursor will change from an arrow to a cross. Pressing any key will send that key and the current coordinate of the cross cursor. Pressing button one, two, or three will return the letters ‘l’, ‘m’, and ‘r’, respectively. If the ‘shift’ key is pressed when a pointer button is pressed, the corresponding upper case letter is sent. To distinguish a pointer button from a key, the high bit of the character is set (but this bit is normally stripped unless the terminal mode is RAW; see `tty(1)` and `tty(7)` for details).

MENUS

Xterm has four menus, named *mainMenu*, *vtMenu*, *fontMenu*, and *tekMenu*. Each menu pops up under the correct combinations of key and button presses. Most menus are divided into two sections, separated by a horizontal line. The top portion contains various modes that can be altered. A check mark appears next to a mode that is currently active. Selecting one of these modes toggles its state. The bottom portion of the menu are command entries; selecting one of these performs the indicated function.

The **xterm** menu pops up when the “control” key and pointer button one are pressed in a window. The *mainMenu* contains items that apply to both the VT102 and Tektronix windows. The **Secure Keyboard** mode is used when typing in passwords or other

sensitive data in an unsecure environment; see **SECURITY** below. Notable entries in the command section of the menu are the **Continue**, **Suspend**, **Interrupt**, **Hangup**, **Terminate** and **Kill** which sends the SIGCONT, SIGTSTP, SIGINT, SIGHUP, SIGTERM and SIGKILL signals, respectively, to the process group of the process running under **xterm** (usually the shell). The **Continue** function is especially useful if the user has accidentally typed CTRL-Z, suspending the process.

The *vtMenu* sets various modes in the VT102 emulation, and is popped up when the “control” key and pointer button two are pressed in the VT102 window. In the command section of this menu, the soft reset entry will reset scroll regions. This can be convenient when some program has left the scroll regions set incorrectly (often a problem when using VMS or TOPS-20). The full reset entry will clear the screen, reset tabs to every eight columns, and reset the terminal modes (such as wrap and smooth scroll) to their initial states just after **xterm** has finished processing the command line options.

The *fontMenu* sets the font used in the VT102 window. In addition to the default font and a number of alternatives that are set with resources, the menu offers the font last specified by the Set Font escape sequence (see the document *Xterm Control Sequences*) and the current selection as a font name (if the PRIMARY selection is owned).

The *tekMenu* sets various modes in the Tektronix emulation, and is popped up when the “control” key and pointer button two are pressed in the Tektronix window. The current font size is checked in the modes section of the menu. The **PAGE** entry in the command section clears the Tektronix window.

SECURITY

X environments differ in their security consciousness. MIT servers, run under *xdm*, are capable of using a “magic cookie” authorization scheme that can provide a reasonable level of security for many people. If your server is only using a host-based mechanism to control access to the server (see **xhost(1)**), then if you enable access for a host and other users are also permitted to run clients on that same host, there is every possibility that someone can run an application that will use the basic services of the X protocol to snoop on your activities, potentially capturing a transcript of everything you type at the keyboard. This is of particular concern when you want to type in a password or other sensitive data. The best solution to this problem is to use a better authorization mechanism that host-based control, but a simple mechanism exists for protecting keyboard input in **xterm**.

The **xterm** menu (see **MENUS** above) contains a **Secure Keyboard** entry which, when enabled, ensures that all keyboard input is directed *only* to **xterm** (using the GrabKeyboard protocol request). When an application prompts you for a password (or other sensitive data), you can enable **Secure Keyboard** using the menu, type in the data, and then disable **Secure Keyboard** using the menu again. Only one X client at a time can secure the keyboard, so when you attempt to enable **Secure Keyboard** it may fail. In this case, the bell will sound. If the **Secure Keyboard** succeeds, the foreground and background colors will be exchanged (as if you selected the **Reverse Video** entry in the **Modes** menu); they will be exchanged again when you exit secure mode. If the colors do *not* switch, then you should be *very* suspicious that you are being spoofed. If the application you are running displays a prompt before asking for the password, it is safest to enter secure

mode *before* the prompt gets displayed, and to make sure that the prompt gets displayed correctly (in the new colors), to minimize the probability of spoofing. You can also bring up the menu again and make sure that a check mark appears next to the entry.

Secure Keyboard mode will be disabled automatically if your xterm window becomes iconified (or otherwise unmapped), or if you start up a reparenting window manager (that places a title bar or other decoration around the window) while in **Secure Keyboard** mode. (This is a feature of the X protocol not easily overcome.) When this happens, the foreground and background colors will be switched back and the bell will sound in warning.

CHARACTER CLASSES

Clicking the middle mouse button twice in rapid succession will cause all characters of the same class (e.g. letters, white space, punctuation) to be selected. Since different people have different preferences for what should be selected (for example, should filenames be selected as a whole or only the separate subnames), the default mapping can be overridden through the use of the *charClass* (class *CharClass*) resource.

This resource is simply a list of *range:value* pairs where the range is either a single number or *low-high* in the range of 0 to 127, corresponding to the ASCII code for the character or characters to be set. The *value* is arbitrary, although the default table uses the character number of the first character occurring in the set.

The default table is:

```
static int charClass[128] = {
  /* NUL SOH STX ETX EOT ENQ ACK BEL */
  32, 1, 1, 1, 1, 1, 1, 1,
  /* BS HT NL VT NP CR SO SI */
  1, 32, 1, 1, 1, 1, 1, 1,
  /* DLE DC1 DC2 DC3 DC4 NAK SYN ETB */
  1, 1, 1, 1, 1, 1, 1, 1,
  /* CAN EM SUB ESC FS GS RS US */
  1, 1, 1, 1, 1, 1, 1, 1,
  /* SP ! " # $ % & ' */
  32, 33, 34, 35, 36, 37, 38, 39,
  /* ( ) * + , - . */
  40, 41, 42, 43, 44, 45, 46, 47,
  /* 0 1 2 3 4 5 6 7 */
  48, 48, 48, 48, 48, 48, 48, 48,
  /* 8 9 : ; < = > ? */
  48, 48, 58, 59, 60, 61, 62, 63,
  /* @ A B C D E F G */
  64, 48, 48, 48, 48, 48, 48, 48,
  /* H I J K L M N O */
  48, 48, 48, 48, 48, 48, 48, 48,
  /* P Q R S T U V W */
  48, 48, 48, 48, 48, 48, 48, 48,
```

```

/* X Y Z [ \ ] ^ _ */
 48, 48, 48, 91, 92, 93, 94, 48,
/* ' a b c d e f g */
 96, 48, 48, 48, 48, 48, 48, 48,
/* h i j k l m n o */
 48, 48, 48, 48, 48, 48, 48, 48,
/* p q r s t u v w */
 48, 48, 48, 48, 48, 48, 48, 48,
/* x y z { | } ~ DEL */
 48, 48, 48, 123, 124, 125, 126, 1};

```

For example, the string “33:48,37:48,45-47:48,64:48” indicates that the exclamation mark, percent sign, dash, period, slash, and ampersand characters should be treated the same way as characters and numbers. This is very useful for cutting and pasting electronic mailing addresses and filenames.

ACTIONS

It is possible to rebind keys (or sequences of keys) to arbitrary strings for input, by changing the translations for the vt100 or tek4014 widgets. Changing the translations for events other than key and button events is not expected, and will cause unpredictable behavior. The following actions are provided for using within the vt100 or tek4014 **translations** resources:

bell(*percent*)

This action rings the keyboard bell at the specified percentage above or below the base volume.

ignore() This action ignores the event but checks for special pointer position escape sequences.

insert() This action inserts the character or string associated with the key that was pressed.

insert-seven-bit()

This action is a synonym for **insert()**

insert-eight-bit()

This action inserts an eight-bit (Meta) version of the character or string associated with the key that was pressed. The exact action depends on the value of the **eightBitInput** resource.

insert-selection(*sourcename* [, ...])

This action inserts the string found in the selection or cutbuffer indicated by *sourcename*. Sources are checked in the order given (case is significant) until one is found. Commonly-used selections include: *PRIMARY*, *SECONDARY*, and *CLIPBOARD*. Cut buffers are typically named *CUT_BUFFER0* through *CUT_BUFFER7*.

keymap(*name*)

This action dynamically defines a new translation table whose resource name is *name* with the suffix *Keymap* (case is significant). The name *None* restores the

original translation table.

popup-menu(*menuname*)

This action displays the specified popup menu. Valid names (case is significant) include: *mainMenu*, *vtMenu*, *fontMenu*, and *tekMenu*.

secure() This action toggles the *Secure Keyboard* mode described in the section named **SECURITY**, and is invoked from the **securekbd** entry in *mainMenu*.

select-start()

This action begins text selection at the current pointer location. See the section on **POINTER USAGE** for information on making selections.

select-extend()

This action tracks the pointer and extends the selection. It should only be bound to Motion events.

select-end(*destname* [, ...])

This action puts the currently selected text into all of the selections or cutbuffers specified by *destname*.

select-cursor-start()

This action is similar to **select-start** except that it begins the selection at the current text cursor position.

select-cursor-end(*destname* [, ...])

This action is similar to **select-end** except that it should be used with **select-cursor-start**.

set-vt-font(*d/1/2/3/4/5/6/e/s* [, *normalfont* [, *boldfont*]])

This action sets the font or fonts currently being used in the VT102 window. The first argument is a single character that specifies the font to be used: *d* or *D* indicate the default font (the font initially used when **xterm** was started), *1* through *6* indicate the fonts specified by the *font1* through *font6* resources, *e* or *E* indicate the normal and bold fonts that have been set through escape codes (or specified as the second and third action arguments, respectively), and *s* or *S* indicate the font selection (as made by programs such as *xfontsel(1)*) indicated by the second action argument.

start-extend()

This action is similar to **select-start** except that the selection is extended to the current pointer location.

start-cursor-extend()

This action is similar to **select-extend** except that the selection is extended to the current text cursor position.

string(*string*)

This action inserts the specified text string as if it had been typed. Quotation is necessary if the string contains whitespace or non-alphanumeric characters. If the string argument begins with the characters "0x", it is interpreted as a hex character constant.

scroll-back(*count* [, *units*])

This action scrolls the text window backward so that text that had previously scrolled off the top of the screen is now visible. The *count* argument indicates the number of *units* (which may be *page*, *halfpage*, *pixel*, or *line*) by which to scroll.

scroll-forw(*count* [,*units*])

This action scrolls is similar to **scroll-back** except that it scrolls the other direction.

allow-send-events(*on/off/toggle*)

This action set or toggles the **allowSendEvents** resource and is also invoked by the **allowsends** entry in *mainMenu*.

set-logging(*on/off/toggle*)

This action toggles the **logging** resource and is also invoked by the **logging** entry in *mainMenu*.

redraw()

This action redraws the window and is also invoked by the *redraw* entry in *mainMenu*.

send-signal(*signame*)

This action sends the signal named by *signame* to the **xterm** subprocess (the shell or program specified with the *-e* command line option) and is also invoked by the **suspend**, **continue**, **interrupt**, **hangup**, **terminate**, and **kill** entries in *mainMenu*. Allowable signal names are (case is not significant): *tstp* (if supported by the operating system), *suspend* (same as *tstp*), *cont* (if supported by the operating system), *int*, *hup*, *term*, *quit*, *alarm*, *alarm* (same as *alarm*) and *kill*.

quit() This action sends a SIGHUP to the subprogram and exits. It is also invoked by the **quit** entry in *mainMenu*.

set-scrollbar(*on/off/toggle*)

This action toggles the **scrollbar** resource and is also invoked by the **scrollbar** entry in *vtMenu*.

set-jumpscroll(*on/off/toggle*)

This action toggles the **jumpscroll** resource and is also invoked by the **jumpscroll** entry in *vtMenu*.

set-reverse-video(*on/off/toggle*)

This action toggles the *reverseVideo* resource and is also invoked by the **reversevideo** entry in *vtMenu*.

set-autowrap(*on/off/toggle*)

This action toggles automatic wrapping of long lines and is also invoked by the **autowrap** entry in *vtMenu*.

set-reversewrap(*on/off/toggle*)

This action toggles the **reverseWrap** resource and is also invoked by the **reversewrap** entry in *vtMenu*.

set-autolinefeed(*on/off/toggle*)

This action toggles automatic insertion of linefeeds and is also invoked by the

autolinefeed entry in *vtMenu*.

set-appcursor(*on/off/toggle*)

This action toggles the handling Application Cursor Key mode and is also invoked by the **Bappcursor** entry in *vtMenu*.

set-appkeypad(*on/off/toggle*)

This action toggles the handling of Application Keypad mode and is also invoked by the **appkeypad** entry in *vtMenu*.

set-scroll-on-key(*on/off/toggle*)

This action toggles the **scrollKey** resource and is also invoked from the **scroll-key** entry in *vtMenu*.

set-scroll-on-tty-output(*on/off/toggle*)

This action toggles the **scrollTtyOutput** resource and is also invoked from the **scrollttyoutput** entry in *vtMenu*.

set-allow132(*on/off/toggle*)

This action toggles the **c132** resource and is also invoked from the **allow132** entry in *vtMenu*.

set-cursesemul(*on/off/toggle*)

This action toggles the **curses** resource and is also invoked from the **cursesemul** entry in *vtMenu*.

set-visual-bell(*on/off/toggle*)

This action toggles the **visualBell** resource and is also invoked by the **visualbell** entry in *vtMenu*.

set-marginbell(*on/off/toggle*)

This action toggles the **marginBell** resource and is also invoked from the **marginbell** entry in *vtMenu*.

set-altscreen(*on/off/toggle*)

This action toggles between the alternate and current screens.

soft-reset()

This action resets the scrolling region and is also invoked from the **softreset** entry in *vtMenu*.

hard-reset()

This action resets the scrolling region, tabs, window size, and cursor keys and clears the screen. It is also invoked from the **hardreset** entry in *vtMenu*.

clear-saved-lines()

This action does **hard-reset()** (see above) and also clears the history of lines saved off the top of the screen. It is also invoked from the **clearsavedlines** entry in *vtMenu*.

set-terminal-type(*type*)

This action directs output to either the *vt* or *tek* windows, according to the *type* string. It is also invoked by the **tekmode** entry in *vtMenu* and the **vtmode** entry in *tekMenu*.

set-visibility(*vt/tek,on/off/toggle*)

This action controls whether or not the *vt* or *tek* windows are visible. It is also invoked from the **tekshow** and **vthide** entries in *vtMenu* and the **vtshow** and **tekhide** entries in *tekMenu*.

set-tek-text(*large/2/3/small*)

This action sets font used in the Tektronix window to the value of the resources **tektextlarge**, **tektext2**, **tektext3**, and **tektextsmall** according to the argument. It is also by the entries of the same names as the resources in *tekMenu*.

tek-page()

This action clears the Tektronix window and is also invoked by the **tekpage** entry in *tekMenu*.

tek-reset()

This action resets the Tektronix window and is also invoked by the *tekreset* entry in *tekMenu*.

tek-copy()

This action copies the escape codes used to generate the current window contents to a file in the current directory beginning with the name COPY. It is also invoked from the *tekcopy* entry in *tekMenu*.

visual-bell()

This action flashes the window quickly.

The Tektronix window also has the following action:

gin-press(*/L/m/M/r/R*)

This action sends the indicated graphics input code.

The default bindings in the VT102 window are:

| | |
|--------------------------|---|
| Shift <KeyPress> Prior: | scroll-back(1,halpage) \n\ |
| Shift <KeyPress> Next: | scroll-forw(1,halpage) \n\ |
| Shift <KeyPress> Select: | select-cursor-start() \ |
| | select-cursor-end(PRIMARY, CUT_BUFFER0) \n\ |
| Shift <KeyPress> Insert: | insert-selection(PRIMARY, CUT_BUFFER0) \n\ |
| ~Meta<KeyPress>: | insert-seven-bit() \n\ |
| Meta<KeyPress>: | insert-eight-bit() \n\ |
| !Ctrl <Btn1Down>: | popup-menu(mainMenu) \n\ |
| !Lock Ctrl <Btn1Down>: | popup-menu(mainMenu) \n\ |
| ~Meta <Btn1Down>: | select-start() \n\ |
| ~Meta <Btn1Motion>: | select-extend() \n\ |
| !Ctrl <Btn2Down>: | popup-menu(vtMenu) \n\ |
| !Lock Ctrl <Btn2Down>: | popup-menu(vtMenu) \n\ |
| ~Ctrl ~Meta <Btn2Down>: | ignore() \n\ |
| ~Ctrl ~Meta <Btn2Up>: | insert-selection(PRIMARY, CUT_BUFFER0) \n\ |
| !Ctrl <Btn3Down>: | popup-menu(fontMenu) \n\ |
| !Lock Ctrl <Btn3Down>: | popup-menu(fontMenu) \n\ |
| ~Ctrl ~Meta <Btn3Down>: | start-extend() \n\ |

```

~Meta <Btn3Motion>:      select-extend() \n\
                        <BtnUp>:select-end(PRIMARY, CUT_BUFFER0) \n\
<BtnDown>:              bell(0)

```

The default bindings in the Tektronix window are:

```

~Meta<KeyPress>:        insert-seven-bit() \n\
Meta<KeyPress>:        insert-eight-bit() \n\
!Ctrl <Btn1Down>:      popup-menu(mainMenu) \n\
!Lock Ctrl <Btn1Down>: popup-menu(mainMenu) \n\
!Ctrl <Btn2Down>:      popup-menu(tekMenu) \n\
!Lock Ctrl <Btn2Down>: popup-menu(tekMenu) \n\
Shift ~Meta<Btn1Down>: gin-press(L) \n\
~Meta<Btn1Down>:       gin-press(l) \n\
Shift ~Meta<Btn2Down>: gin-press(M) \n\
~Meta<Btn2Down>:       gin-press(m) \n\
Shift ~Meta<Btn3Down>: gin-press(R) \n\
~Meta<Btn3Down>:       gin-press(r)

```

Below is a sample how of the **keymap0** action is used to add special keys for entering commonly-typed works:

```

*VT100.Translations: #override <Key>F13: keymap(dbx)
*VT100.dbxKeymap.translations: \
    <Key>F14:      keymap(None) \n\
    <Key>F17:      string("next") string(0x0d) \n\
    <Key>F18:      string("step") string(0x0d) \n\
    <Key>F19:      string("continue") string(0x0d) \n\
    <Key>F20:      string("print ") insert-selection(PRIMARY, CUT_BUFFER0)

```

ENVIRONMENT

Xterm sets the environment variable “TERM” properly for the size window you have created. It also uses and sets the environment variable “DISPLAY” to specify which bit map display terminal to use. The environment variable “WINDOWID” is set to the X window id number of the **xterm** window.

SEE ALSO

resize(1), **X11(7)**, **tty(1)**
Xterm Control Sequences (in the **xterm** source directory)

BUGS

Large pastes do not work on some systems. This is not a bug in **xterm**; it is a bug in the pseudo terminal driver of those systems. **xterm feeds large pastes to the** will accept data, but some pty drivers do not return enough information to know if the write has succeeded.

Many of the options are not resettable after **xterm** starts.

The Tek widget does not support key/button re-binding.

Only fixed-width, character-cell fonts are supported.

This program still needs to be rewritten. It should be split into very modular sections, with the various emulators being completely separate widgets that don't know about each other. Ideally, you'd like to be able to pick and choose emulator widgets and stick them into a single control widget.

There needs to be a dialog box to allow entry of log file name and the COPY file name.

COPYRIGHT

Copyright 1989, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHORS

Far too many people, including:

Loretta Guarino Reid (DEC-UEG-WSL), Joel McCormack (DEC-UEG-WSL), Terry Weissman (DEC-UEG-WSL), Edward Moy (Berkeley), Ralph R. Swick (MIT-Athena), Mark Vandevoorde (MIT-Athena), Bob McNamara (DEC-MAD), Jim Gettys (MIT-Athena), Bob Scheifler (MIT X Consortium), Doug Mink (SAO), Steve Pitschke (Stellar), Ron Newman (MIT-Athena), Jim Fulton (MIT X Consortium), Dave Serisky (HP), Jonathan Kamens (MIT-Athena)

| | |
|--|--|
| NAME | xview – xview toolkit information |
| SYNOPSIS | There is no xview command per se, but this manual page will briefly describe XView features and functions. |
| AVAILABILITY | XView is available with the OpenWindows distribution. |
| DESCRIPTION | XView (X Window-System-based Visual/Integrated Environment for Workstations) is an Open Look user-interface toolkit which supports development of interactive, graphics-based applications running under the X Window System. For detailed information see the <i>XView Programming Manual</i> and the <i>XView Reference Manual</i> . |
| USAGE Compiling XView Programs | XView programs are compiled with the following command line: cc sourcefile.c -o outputfile -lxview -lolgx -lX11 |
| Generic XView Functions | <p>xv_init() Establishes the connection to the server, initializes the Notifier and the Defaults/Resource-Manager database, loads the Server Resource-Manager database, reads any passed attributes, and installs a default X11 Errorhandler.</p> <p>Xv_Server xv_init(attrs) <attribute-value list> attrs;</p> <p>Note that attrs is a NULL terminated attribute-value list.</p> <p>xv_create() Creates an object.</p> <p>Xv_object xv_create(owner, package, attrs) Xv_object owner; Xv_pkg package; <attribute-value list> attrs;</p> <p>xv_destroy() Destroys an object.</p> <p>int xv_destroy(object) Xv_opaque object;</p> <p>xv_find() Finds an object that meets certain criteria; or if the object doesn't exist,</p> |

creates it (default behavior which can be defeated using `XV_AUTO_CREATE, FALSE`).

Xv_opaque

```
xv_find(owner, package, attrs)
  Xv_object  owner;
  Xv_pkg     package;
  <attribute-value list> attrs;
```

xv_get() Gets the value of a single attribute.

Xv_opaque

```
xv_get(object, attrs)
  Xv_object  object;
  <attribute-value list> attrs;
```

xv_set() Sets the value of one or more attributes.

Xv_opaque

```
xv_set(object, attrs)
  Xv_object  object;
  <attribute-value list> attrs;
```

**Internationalized
Support**

XView now has support for internationalization. This includes locale setting, localized text handling, and dynamic object layout. See the *XView Programming Manual* for details.

**Command Line
Resource Arguments**

XView-based applications display characteristics can be controlled by supplying command line arguments to the applications at start-up. The usage is as follows:

```
% program -argument1 value1 -argument2 value2...
```

In the tables below, **Argument(s)** shows the short argument followed by the long argument—either can be used. **Type** describes the type of value the arguments can receive. **Resource** describes the X resource name modified by the arguments. **Default** is the default value. **Description** describes what the arguments do. **Example** shows an example of a command using the argument.

| | |
|---------------------|--|
| Argument(s): | -Wx, or -scale |
| Type: | string ("small", "medium", "large", "extra_large") |
| Resource: | Window.Scale |
| Default: | medium |
| Description: | Sets the initial scale of the application (larger or smaller). small is 10 pixels, medium is 12 pixels, large is 14 pixels and extra_large is 19 pixels. The font.name resource will override the scale. |
| Example: | cmdtool -scale extra_large |

| | |
|---------------------|---|
| Argument(s): | -Wt, -fn, or -font |
| Type: | string |
| Resource: | Font.Name |
| Default: | lucidasans-12 |
| Description: | Sets the name of the font used for the application. Does not set the font for frame header and frame menu header. These are controlled by the window manager. To find out what fonts are available, use the xlsfonts(1) command. If the font you specify cannot be found, you will see an error message such as: <i>XView warning: Cannot load font set 'galant-24' (Font package)</i> |
| Example: | cmdtool -fn fixed |
| Argument(s): | -Ws, or -size |
| Type: | integer integer |
| Resource: | Window.Width and Window.Height |
| Default: | depends on application |
| Description: | Sets the width and height of the application's base frame. The values are in pixels. |
| Example: | cmdtool -Ws 400 500 |
| Argument(s): | -Ww, or -width |
| Type: | int (number of columns) |
| Resource: | window.columns |
| Default: | None |
| Description: | Specifies the width, in columns, of the application. |
| Example: | cmdtool -width 40 (starts a command tool 40 columns wide) |
| Argument(s): | -Wh, or -height |
| Type: | int (number of columns) |
| Resource: | window.rows |
| Default: | None |
| Description: | Specifies the height, in rows, of the application. |
| Example: | cmdtool -height 40 (starts a command tool 40 rows high) |
| Argument(s): | -Wp, or -position |
| Type: | integer integer |
| Resource: | Window.X and Window.Y |
| Default: | depends on window manager |
| Description: | Sets the initial position of the application's base frame in pixels. The upper left corner of the screen is at position (0,0), with the x-axis increasing to the left, and the y-axis increasing downward. These values will also be generated by the "Save Workspace" option on the root menu into the \$HOME/.openwin-init file when using the Open Look Window |

Manager.

Example: `cmdtool -Wp 100 200`

Argument(s): -WG, or -geometry
Type: string of the format <width>x<height>{+-}<xoffset>{+-}<yoffset>
Resource: Window.Geometry
Default: depends on application and window manager
Description: This sets both the size and the placement of the application's base frame. This option has priority over the **-size** and **-position** arguments. The size and placement parts of the value are optional. You can set just the size, just the position, or both. The size values are measured in pixels, and the position values use the same semantics as **-position**. However, if you use the '-' in front of an X value, it will be taken as relative to the right hand side of the screen, instead of the left. Likewise, if you use the '-' with the Y value, it will be taken relative to the bottom of the screen instead of the top.

Examples: `cmdtool -geometry 500x600`
(will make the base frame 500x600 pixels, with the position set by the window manager)
`cmdtool -WG +10+20`
(will make the base frame of default size with the left hand side of the frame 10 pixels from the left hand side of the screen, and the top of the frame 20 pixels from the top of the screen)
`cmdtool -WG -10+20`
(will make the base frame of default size with the right hand side of the frame 10 pixels from the right hand side of the screen, and the top of the frame 20 pixels from the top of the screen)
`cmdtool -geometry 400x300-0-0`
(will make the base frame 400x300 pixels with the right hand side of the frame flush against the right hand side of the screen, and the bottom of the frame flush with the bottom of the screen)

Argument(s): -WP, -icon_position
Type: integer integer
Resource: Icon.X Icon.Y
Default: depends on window manager
Description: Sets the position of the application's icon in pixels. Uses the same semantics as **-position** for base frames.

Example: `cmdtool -WP 400 20`

Argument(s): -Wl, -label, or -title
Type: string
Resource: Window.Header
Default: N/A
Description: Sets a default label for the base frame's header. However, the

application can overwrite this setting and display its own header.

Example: `cmdtool -WI "Header Text"`

Argument(s): -Wi, and +Wi
Type: boolean
Resource: Window.Iconic
Default: +Wi
Description: Controls how an application will come up, open or closed (iconified).
Examples: `cmdtool +Wi` (will make the cmdtool come up open) `cmdtool -Wi` (will make the cmdtool come up closed)

Argument(s): -depth
Type: integer
Resource: Window.Depth
Default: Depth of server's default visual
Description: Specifies the depth of base frame. If this depth is not supported by the server, the default depth will be used instead. If this is specified in conjunction with -visual, then the exact visual will be used.
Example: `cmdtool -depth 4`

Argument(s): -visual
Type: string (one of the values: StaticGray, GrayScale, StaticColor, PseudoColor, TrueColor, or DirectColor).
Resource: Window.Visual
Default: Server's default visual
Description: Specifies the visual class of the base frame. If this visual class is not supported by the server, the default visual class will be used instead. If this is specified in conjunction with -depth, then the exact visual will be used.
Example: `cmdtool -visual StaticGray`

Argument(s): -Wf, or -foreground_color
Type: integer integer integer
Resource: Window.Color.Foreground
Default: 0 0 0
Description: See Description in -Wb below.

Argument(s): -Wb, or -background
Type: integer integer integer
Resource: Window.Color.Background
Default: 255 255 255
Description: These options allow the user to specify the foreground color (e.g., the color of the text in a textsw), or the background color (e.g., the color that the text is painted on) of an application. The three values should be integers between 0 and 255. They specify the amount of red, green and

blue that is in the color. See **-fg** and **-bg** below for information on similar functions.

Example: **cmdtool -Wf 0 0 255 -Wb**
(would come up with a blue foreground, with a gray background)

Argument(s): -fg, or -foreground
Type: string (color name, or hexadecimal color specification)
Resource: Window.Color.Foreground
Default: black
Description: See Description in -bg below.

Argument(s): -bg, or -background
Type: string (color name, or hexadecimal color specification)
Resource: Window.Color.Background
Default: white
Description: These options are similar to the -Wf and -Wb options, except that they take a color argument in the form of a predefined color name (lavender, grey, goldenrod, etc.) from `$OPENWINHOME/lib/rbg.txt`, or a hexadecimal representation. The hexadecimal representation is of the form pound sign (#) followed by the hexadecimal representation of the red, green and blue aspects of the color.

Examples: **cmdtool -fg blue -bg gray**
(comes up with a blue foreground, with a gray background)
cmdtool -fg #d800ff -bg white
(comes up with a purple foreground, with a white background)

Argument(s): -WI, or -icon_image
Type: string
Resource: Icon.Pixmap
Default: depends on application
Description: Sets the default filename for the icon's image. However, the application can overwrite this setting and display its own icon image. The file must be in XView icon format. The program `iconedit(1)` will allow one to create an image in the icon format. Several icons have been provided in the directory `$OPENWINHOME/include/images`. By convention, icon format files end with the suffix `.icon`.

Example: **cmdtool -WI /usr/include/images/stop.icon**

Argument(s): -WL, or -icon_label
Type: string
Resource: Icon.Footer
Default: depends on application
Description: Sets a default label for the base frame's icon. However, the application can overwrite this setting and display its own icon label.

Example: **cmdtool -WL "Icon Label"**

| | |
|---------------------|--|
| Argument(s): | -WT, or -icon_font |
| Type: | string |
| Resource: | Icon.Font.Name |
| Default: | depends |
| Description: | Sets the name of the font used for the application's icon. |
| Example: | cmdtool -WT '*century schoolbook*' |
| Argument(s): | -Wd, or -default |
| Type: | string string |
| Resource: | given by the first string |
| Default: | none |
| Description: | This option allows the user to set resources that don't have command line equivalents. The format is -default resource-name value . The XView resources without specific command line arguments are discussed in the following section. |
| Example: | cmdtool -default OpenWindows.ScrollbarPlacement left |
| Argument(s): | -xrm |
| Type: | string |
| Resource: | given in the string |
| Default: | none |
| Description: | This option allows the user to set resources that don't have command line equivalents. This is similar to the -default option, but it takes only one argument, a string in the form of resource-name:value. |
| Example: | cmdtool -xrm OpenWindows.ScrollbarPlacement:right |
| Argument(s): | -WH, or -help |
| Type: | none |
| Resource: | none |
| Default: | none |
| Description: | Prints a description of the valid xview command line arguments for the application. |
| Argument(s): | -sync or -synchronous, and +sync or +synchronous |
| Type: | boolean |
| Resource: | Window.Synchronous |
| Default: | +synchronous |
| Description: | These options allow you to make the connection that the application has with the X11 server either synchronous (-sync) or asynchronous (+sync). |
| Argument(s): | -Wr, or -display |
| Type: | string (host:display{.screen}) |
| Resource: | Server.Name |
| Default: | taken from the DISPLAY environment variable |
| Description: | Sets the name of the X11 server on which to connect. host is the name or |

address of the machine on whose server you have permission to display. **display** is a number corresponding to the server on which to display for that machine, and **screen** corresponds to which screen for the server. See reference manual page on **xhost(1)** for more details on adding to permissions list.

- Examples:** **cmdtool -display foobar:0**
(will bring up a cmdtool on the default screen of the display #0 of host foobar)
cmdtool -display foobar:0.1
(will bring up a cmdtool on screen #1 of display #0 of host foobar)
- Argument(s):** -Wdxio, or -disable_xio_error_handler
Type: boolean
Resource: none
Default: enable xio handler—this option disables it
Description: This option is useful for debugging an application. Whenever there is a fatal XIO error, the server will print an error message before exiting. XView installs a error handler to keep those messages from appearing. If you would like to see these messages, use this option.
- Argument(s)** -rv (or -reverse), and +rv (or +reverse)
Type: boolean
Resource: Window.ReverseVideo
Default: False
Description: These options control whether the foreground and background colors of the application will be reversed. If True, the foreground and background colors will be swapped. The -rv flag will set this to True, while the +rv will set it to False. This is really only useful on monochrome displays.
- Argument(s):** -name
Type: string
Resource: None
Default: argv[0]
Description: Specifies the instance name of the application. This name is used to construct the resource name used to perform lookups in the X11 Resource Manager to look for the values of customizable attributes.

Internationalized Command Line Resource Arguments

The following command line arguments are relevant to internationalization. Locale refers to the language and cultural conventions used in a program. Locale setting is the method by which the language and cultural environment of a system is set. Locale setting affects the display and manipulation of language-dependent features.

The internationalization features that XView now supports include locale setting. One of the ways locale can be set is with command line options. See the *XView Programming*

Manual for details on other methods.

Argument(s): -lc_basicalocale
Type: string
Resource: basicLocale
Default: "C"
Description: Specifies the basic locale category, which sets the country of the user interface.

Argument(s): -lc_displaylang
Type: string
Resource: displayLang
Default: "C"
Description: Specifies the display language locale category, sets the language in which labels, messages, menu items, and help text are displayed.

Argument(s): -lc_inputlang
Type: string
Resource: inputLang
Default: "C"
Description: Specifies the input language locale category, sets the language used for keyboard input.

Argument(s): -lc_numeric
Type: string
Resource: numeric
Default: "C"
Description: Specifies the numeric locale category, which defines the language used to format numeric quantities.

Argument(s): -lc_timeformat
Type: string
Resource: timeFormat
Default: "C"
Description: Specifies the time format locale category, which defines the language used to format time and date.

Command Line Options/X Resources for Debugging

The following switches/resources can be used during development to avoid the locking up of screens or other effects of X grabs that are done by XView.

It should be noted that these options/resources should only be used by developers and are not for normal usage. The X grabs are done for a specific reason, and are not meant to be customizable. Without the X grabs, certain features in XView (those that depend on X grabs) might not function properly.

Argument(s): -Wfsdb, or -fullscreendebug
Type: boolean
Resource: Fullscreen.Debug
Default: FALSE
Description: Enables/disables fullscreen debugging mode during which X grabs (XGrabServer(), XGrabKeyboard(), XGrabPointer()) are not done. When using the fullscreen pkg, the X11 server will be grabbed which prevents other windows on the server from responding until the grab has been released by the one window which initiated the grab.

Argument(s): -Wfsdbs, or -fullscreendebugserver
Type: boolean
Resource: Fullscreen.Debugserver
Default: FALSE
Description: Enables/disables server grabbing (XGrabServer()) that is done via the fullscreen pkg.

Argument(s): -Wfsdbk, or -fullscreendebugkbd
Type: boolean
Resource: Fullscreen.Debugkbd
Default: FALSE
Description: Enables/disables keyboard grabbing (XGrabKeyboard()) that is done via the fullscreen pkg.

Argument(s): -Wfsdbp, or -fullscreendebugptr
Type: boolean
Resource: Fullscreen.Debugptr
Default: FALSE
Description: Enables/disables pointer grabbing (XGrabPointer()) that is done via the fullscreen pkg.

Argument(s): -Wdpgs, or -disable_pass_grab_select
Type: boolean
Resource: Window.PassiveGrab.Select
Default: TRUE
Description: Disables the passive grab that is done on the SELECT button. XView does a passive grab on the SELECT button to avoid input focus race conditions. When this passive grab is disabled, input focus race conditions may be seen.

Example: % cmdtool -disable_pass_grab_select
 This executes a cmdtool that does not perform any passive grabs on the SELECT button. To do the same thing using X resources, add the following entry to the X resource database:
 Window.PassiveGrab.Select:False

.Xdefaults File

The **.Xdefaults** file is used to store and retrieve resource settings. We recommend, however, that you use the command line arguments described above in order to change display characteristics. Changing the resources in the **.Xdefaults** file will modify the behaviour of the user's session. Novice users should not casually hand modify these settings. Before attempting edits to this file please read the appropriate sections of the *Xlib Programming Manual* on the file format and the specific properties you intend to change.

Note that resources documented below do not have command line arguments. It is still possible, however, to change them without altering the **.Xdefaults** file. Refer to the command line arguments **-xrm** and **-defaults** for instructions on how to do this. Additional resources that have command line arguments are documented in the previous section. For mouseless resources refer to the *XView Programming Manual*.

The resources are documented in the following format:

| | |
|--------------------|--|
| Resource: | <i>Resource Name</i> (If the resource can be modified by the OpenWindows Property Sheet, the word Props will be present.) |
| Values: | <i>Possible Values, and/or Format of Values to be Assigned to Resource (Default Value)</i> |
| Description | <i>Description of Resource.</i> |
| Resource: | window.synchronous, +sync -sync |
| Values: | True, False (False) |
| Description | Useful when debugging or tracking down a problem since the error codes emitted from Xlib will correspond to the immediate request made. Running in a synchronous mode will cause the application to run significantly slower. |
| Resource: | mouse.modifier.button2 |
| Values: | Shift, Ctrl, any valid modifier keysym (Shift) |
| Description | When using a mouse with less than three buttons, this resource gets an equivalent mapping for the second button which is the ADJUST button on a three button mouse. For more information on keysyms, see the xmodmap (1) reference manual page, Xlib documentation, and the include file \$OPENWINHOME/include/X11/Xkeymap.h . |
| Resource: | mouse.modifier.button3 |
| Values: | Shift, Ctrl, any valid modifier keysym (Ctrl) |
| Description | When using a mouse with less than three buttons, this resource gets an equivalent mapping for the third button which is the MENU button on a three button mouse. For more information on keysyms, see the xmodmap reference manual page, Xlib documentation, and the include file \$OPENWINHOME/include/X11/Xkeymap.h . |
| Resource: | OpenWindows.beep (Props) |

| | |
|--|--|
| Values: | never, notices, always (always) |
| Description | When the value is notices , the audible bell will ring only when a notice pops up. When the value is never , the audible bell will never ring. When the value is always , the audible bell will always ring when the bell function is called by a program. |
| Resource: | alarm.visible |
| Values: | True, False (True) |
| Description | When ringing the bell in an XView program, flash the window as well to alert the user. |
| OpenWindows.windowColor (Props) | |
| Values: | any valid X11 color specification (#cccccc—80% grey) |
| Description | Specify the base color for control areas for 3-D look. Takes hexadecimal representation. Three other colors used for shading and highlighting are calculated based upon the value of the specified control color. The actual calculated values are done by the OLGX library to provide a consistent color calculation between XView and OLWM. The desktop properties program allows a full range of customization and previews what the chosen 3-D look will look like. Does not apply to monochrome displays. |
| Resource: | OpenWindows.workspaceColor (Props) |
| Values: | any valid X11 color specification (#cccccc—80% grey) |
| Description | Specifies the color for the root window and the background color for icons that blend into the desktop. |
| Resource: | xview.iccmmcompliant |
| Values: | True, False (True) |
| Description | When False, tells XView to set window manager hints in a way that was used before the ICCCM was adopted. Useful for window managers that were released before X11R4. Not needed with the Open Look Window Manager provided with Open Windows. |
| Resource: | OpenWindows.3DLook.Color |
| Values: | True, False (True on all but monochrome screens) |
| Description | When False, do not use the 3-D look on a color or greyscale screen. |
| Resource: | OpenWindows.dragRightDistance (Props) |
| Values: | N (100) |
| Description | Used by menus to determine when a pullright submenu would display when dragging over the menu item near a submenu. N is an integer greater than 0. A reasonable value might start at 20 and go to 200 or so. May need to try different values to see what feels best to each person. |

| | |
|---------------------|---|
| Resource: | Selection.Timeout |
| Values: | N (3) |
| Description: | Selection timeout value. N indicates the number of seconds that a requestor or a selection owner waits for a response. |
| Resource: | OpenWindows.GotoMenu.RecentCount |
| Values: | integer between 0 and 15 (8) |
| Description: | Specifies the number of recently visited directories shown in the Go To Menu of a File Chooser. |
| Resource: | OpenWindows.GotoMenu.UserDirs |
| Values: | string-list (NULL) |
| Description: | new-line (0 separated list of full-path names to directories that is added to the top of the Go To Menu of a File Chooser. |
| Resource: | OpenWindows.KeyboardCommand.* |
| Description: | These resources determine mouseless semantic action and its corresponding key binding. Refer to the <i>XView Reference Manual</i> for a complete listing and explanation of the OpenWindows.KeyboardCommand.* resources. Refer to the <i>XView Programming Manual</i> for information on the mouseless model. |
| Resource: | OpenWindows.KeyboardCommands |
| Values: | SunView1, Basic, or Full |
| Description: | Controls the level of mouseless operation. All of the OpenWindows.KeyboardCommand resource mappings may be modified by users, or by specifying one the the three values for OpenWindows.KeyboardCommands. For detailed information see the <i>XView Programming Manual</i> |
| Resource: | OpenWindows.MenuAccelerators |
| Values: | True or False (True) |
| Description: | Specifies whether or not to activate all keyboard menu acceleration defined by applications. Menu accelerators are keystrokes that can be used to invoke menu commands directly. They can be seen on the right side of frequently used menu items as a set of keyboard qualifiers (with a graphical diamond mark representing the meta key) and an accelerator key. |
| Resource: | OpenWindows.MouseChordMenu |
| Values: | True, False (False) |
| Description: | Turns on the mouse chording mechanism. Mouse chording was implemented to allow XView to work with two-button mice. Pressing the SELECT and the ADJUST buttons at the same time will act as MENU button. |

| | |
|---------------------|---|
| Resource: | OpenWindows.MouseChordTimeout |
| Values: | N (100) |
| Description: | Mouse chording time-out value. N is in micro-seconds. |
| Resource: | OpenWindows.SelectDisplaysMenu (Props) |
| Values: | True, False (False) |
| Description: | When True, the SELECT button (usually left mouse) will display the menu as well as the MENU button (usually right mouse). |
| Resource: | OpenWindows.popupJumpCursor (Props) |
| Values: | True, False (False) |
| Description: | When False, do not warp the mouse to the notice when it appears. |
| Resource: | notice.beepCount |
| Values: | N (1) |
| Description: | Where N is an integer to specify how many times to ring the bell when a notice appears. |
| Resource: | OpenWindows.scrollbarPlacement (Props) |
| Values: | Left, Right (Right) |
| Description: | When set to Left , put all scrollbars on the lefthand side of the window or object. |
| Resource: | OpenWindows.multiClickTimeout (Props) |
| Values: | N (4) |
| Description: | Where N is an integer between 2 and 10, inclusive. Set the number of tenths of a second between clicks for a multi-click. A click is button-down, button-up pair. |
| Resource: | text.delimiterChars |
| Values: | string (' \011!\"#\$%&\'()*+,-./:;<=>?@[\\]_`{ }~') |
| Description: | This resource allows the user to select the delimiter characters that are used when doing word level selections in the XView package. It was added because of the needs of the international marketplace, and it allows the user to define the local delimiters for the character set that is being used with the current keyboard and Sun workstation. |

Note that the octal characters can be scrambled by **Xrm** during a rewrite of the value of text.delimiter.Chars. **Xrm** interprets the text.delimiterChar string when it is loaded. Specifically it will decode the backslashed portions of the string and convert them to octal representations. When this is passed to the client application, the logic will function correctly. However, this misbehavior of **Xrm** causes the string to be stored incorrectly if the user saves the **.Xdefaults** file using the **Xrm** content of the string. The specific problem(s) that occur are the

stripping of the backslash characters and the expansion of the tab character (`\011`).

To correct this problem, one can put the `text.delimiterChar` entry into an `.Xdefaults` file that will not be overwritten when saving the workspace properties (for example, a system wide defaults file). Or a copy of the `text.delimiterChar` entry can be inserted after `.Xdefaults` file saves.

| | |
|---------------------|--|
| Resource: | <code>scrollbar.jumpCursor</code> (Props) |
| Values: | True, False (True) |
| Description: | When False, the scrollbar will not move the mouse pointer when scrolling. |
| Resource: | <code>scrollbar.repeatDelay</code> |
| Values: | N (100) |
| Description: | Where N is some integer greater than 2. Specifies the time in milliseconds when a click becomes a repeated action. |
| Resource: | <code>scrollbar.pageInterval</code> |
| Values: | N (100) |
| Description: | Where N is some integer greater than 2. Specifies the time in milliseconds between repeats of a single page scroll. |
| Resource: | <code>scrollbar.lineInterval</code> |
| Values: | N (1) |
| Description: | Where N is some integer greater than 0. Specifies the time in milliseconds between repeats of a single line scroll. How long to pause scrolling when holding down the SELECT button on the scrollbar elevator. Scrollbar sets up a timer routine for repeats. |
| Resource: | <code>text.maxDocumentSize</code> |
| Values: | N (2000) |
| Description: | Where N specifies the bytes used in memory before a text file is saved to a file on disk. Once this limit is exceeded, the text package will send a notice to the user to tell them that no more insertions are possible. If the file being edited is saved to a file, or it is a disk file being edited, then the limit does not apply. |
| Resource: | <code>text.retained</code> |
| Values: | True, False (False) |
| Description: | If True, retain text windows with server backing store. |
| Resource: | <code>text.extrasMenuFilename</code> |
| Values: | filename (<code>/usr/lib/.text_extras_menu</code>) |
| Description: | Where filename is an absolute location to a file. Can also be set via |

environment variable EXTRASMENU. This file is used for the text package's Extras menu. The commands specified in the extras menu are applied to the contents of the current selection in the textsw window and then it inserts the results at the current insertion point.

| | |
|--------------------|---|
| Resource: | text.enableScrollbar |
| Values: | True, False (True) |
| Description | When False, do not put a scrollbar on textsw objects. |
| Resource: | text.againLimit |
| Values: | N (1) |
| Description | Where N is an integer between 0 and 500. Number of operations the "again history" remembers for a textsw. |
| Resource: | text.autoIndent |
| Values: | True, False (False) |
| Description | When True, begin the next line at the same indentation as the previous line as typing in text. |
| Resource: | text.autoScrollBy |
| Values: | N (1) |
| Description | Where N is an integer between 0 and 100. Specifies the number of lines to scroll when type-in moves insertion point below the view. |
| Resource: | text.confirmOverwrite |
| Values: | True, False (True) |
| Description | When False, do not give user confirmation if a save will overwrite an existing file. |
| Resource: | text.displayControlChars |
| Values: | True, False (True) |
| Description | When False, use an up arrow plus a letter to display the control character instead of the character that is available for the current font. |
| Resource: | Text.DeleteReplacesClipboard |
| Values: | True, False (False) |
| Description | This resource controls whether text that has been selected and then deleted by the delete key or replaced by any other keystroke will be copied to the clipboard. If the value is True, then the selected text will be copied to the clipboard. If the value is False, then the text selected does not replace the clipboard. |
| | This resource also applies to the text selected for the filter function. If the resource is True, then the text selected for a filter function will replace the clipboard when the filter successfully finishes. If the |

resource is False, then the text selected does not replace the clipboard.

| | |
|---------------------|---|
| Resource: | text.undoLimit |
| Values: | N (50 maximum of 500) |
| Description: | Where N is an integer between 0 and 500. How many operations to save in the undo history log. These operations will be undone when you press the "Undo" key in the text window. |
| Resource: | text.insertMakesCaretVisible |
| Values: | If_auto_scroll (Always) |
| Description: | Controls whether insertion causes repositioning to make inserted text visible. |
| Resource: | text.lineBreak |
| Values: | Clip, Wrap_char, Wrap_word (Wrap_word) |
| Description: | Determines how the textsw treats file lines when they are too big to fit on one display line. |
| Resource: | text.margin.bottom |
| Values: | N (0) |
| Description: | Where N is an integer between -1 and 50. Specifies the minimum number of lines to maintain between insertion point and bottom of view. A value of -1 turns auto scrolling off. |
| Resource: | mouse.multiclick.space |
| Values: | N (4) |
| Description: | Where N is an integer between 2 and 500. Specifies the maximum number of pixels between successive mouse clicks to still have the clicks considered as a multi-click event. |
| Resource: | text.storeChangesFile |
| Values: | True, False (True) |
| Description: | When False, do not change the name of the current file being edited to the name of the file that is stored. The name of the current file is reflected in the titlebar of the textedit frame. |
| Resource: | text.margin.top |
| Values: | N (2) |
| Description: | Where N is an integer between -1 and 50. Specifies the minimum number of lines to maintain between the start of the selection and the top of the view. A value of -1 means defeat normal actions. |
| Resource: | text.margin.left |
| Values: | N (8) |
| Description: | Where N is an integer between 0 and 2000. Specifies the margin in |

pixels that the text should maintain between the left hand border of the window and the first character on each line.

Resource: text.margin.right

Values: N (0)

Description: Where N is an integer between 0 and 2000. Specifies the margin in pixels that the text should maintain between the right hand border of the window and the last character on each line.

Resource: text.tabWidth

Values: N (8)

Description: Where N is an integer between 0 and 50. Specifies the width in characters of the tab character.

Resource: Text.LineSpacing

Values: N (0)

Description: Where N is an integer which is the percentage of the maximum height of a character in the Textsw window font to use as interline spacing. Setting Text.LineSpacing to a nonzero positive number will increase the size of a Textsw proportionally. xv_set() of WIN_ROWS will still yield the correct number of rows. However, the window will be taller as compared to a Textsw with Text.LineSpacing set to 0. This resource allows XView to conform to TUV requirements. To meet TUV requirements, set Text.LineSpacing to 15 or greater.

Resource: term.boldStyle

Values: None, Offset_X, Offset_Y, Offset_X_and_Y, Offset_XY, Offset_X_and_XY, Offset_Y_and_XY, Offset_X_and_Y_and_XY, Invert (Invert)

Description: Specify the text bolding style for a terminal based window.

Resource: term.inverseStyle

Values: Enable, Disable, Same_as_bold (Enable)

Description: Specify the text inverting style for a terminal based window.

Resource: term.underlineStyle

Values: Enable, Disable, Same_as_bold (Enable)

Description: Specify the text underlining style for a terminal based window.

Resource: term.useAlternateTtyswrc

Values: True, False (True)

Description: When True, and a \$HOME/.ttyswrc is not found, look for an alternate ttyswrc file. When False, do not look for an alternate file is one is not found in the home directory, \$HOME/.ttyswrc.

| | |
|--------------------|--|
| Resource: | term.alternateTtyswrc |
| Values: | filename (\$XVIEWHOME/lib/.ttyswrc) |
| Description | Where filename specifies a complete filename and absolute path of an alternate ttyswrc file. This is only used if a .ttyswrc file is not found in \$HOME/.ttyswrc and term.useAlternateTtyswrc is True. |
| Resource: | term.enableEdit |
| Values: | True, False (True) |
| Description | When False, do not keep an editlog of what has been typed into the term window. This is set to false automatically when switching from a scrollable term to one that is not scrollable. |
| Resource: | ttysw.eightBitOutput |
| Values: | True, False (True) |
| Description | This resource controls whether characters modified by the meta modifier are encoded as eight-bit characters when passed to the ttysw's pty or are delivered as seven-bit characters. |
| Resource: | ttysw.yieldModifiers |
| Values: | Meta, Alt (The default is to not remove any semantic meaning from any modifiers) |
| Description | This resource takes as a value a list of modifier keys. Any semantic meaning (mouseless command or keyboard accelerator) that would normally be associated with the listed modifiers when the keyboard focus is in a ttysw or termsw would be removed. |

ENVIRONMENT

\$OPENWINHOME is the directory in which the server's directory hierarchy is installed.
\$DISPLAY is the name of the server and screen to which applications should display.
\$LD_LIBRARY_PATH is the SunOS shared library search path.
\$HELPPATH is the path that applications will search for Open Look Help files.

FILES

\$OPENWINHOME/include/images
XView images

\$OPENWINHOME/lib
XView Libraries

\$OPENWINHOME/include
Include files

\$OPENWINHOME/bin
Binaries

\$OPENWINHOME/share/src/xview/demos
XView demo programs

\$OPENWINHOME/share/src/xview/examples
XView example programs

SEE ALSO **openwin(1), props(1), Xsun(1), xlsfonts(1), xmodmap(1), iconedit(1)**

| | |
|--------------------|--|
| NAME | xv_get_sel – copy the content of a selection to the standard output |
| SYNOPSIS | xv_get_sel [rank] [-t seconds] [D] |
| DESCRIPTION | xv_get_sel prints the contents of the indicated selection on standard out. A selection is a collection of objects (for instance, characters) selected with the mouse in the OpenWindow window system. |
| OPTIONS | rank Indicate which selection is to be printed: 1- primary; 2- secondary; 3- clipboard. The default is primary. seconds Indicate how many <i>seconds</i> to wait for the holder of a selection to respond to a request before giving up. The default is six seconds. -D Debugging. Inquire through a special debugging service for the selection, rather than accessing the standard service. Useful only for debugging window applications which are clients of the selection library. |
| EXAMPLES | The following line prints the primary selection on the user's default printer: sh -c xv_get_sel lp |
| SEE ALSO | <i>OpenWindows documentation</i> |

| | |
|--------------------|--|
| NAME | xwd – dump an image of an X window |
| SYNOPSIS | xwd [-debug] [-help] [-nobdrs] [-out <i>file</i>] [-xy] [-frame] [-add <i>value</i>] [-root -id <i>id</i> -name <i>name</i>] [-icmap] [-screen] [-display <i>display</i>] |
| DESCRIPTION | Xwd is an X Window System window dumping utility. Xwd allows X users to store 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. The target window is selected by clicking the pointer in the desired window. The keyboard bell is rung once at the beginning of the dump and twice when the dump is completed. |
| OPTIONS | <p>-display <i>display</i> This argument allows you to specify the server to connect to; see X(7)</p> <p>-help Print out the ‘Usage:’ command syntax summary.</p> <p>-nobdrs This argument specifies that the window dump should not include the pixels that compose the X window border. This is useful in situations where you may wish to include the window contents in a document as an illustration.</p> <p>-out <i>file</i> This argument allows the user to explicitly specify the output file on the command line. The default is to output to standard out.</p> <p>-xy This option applies to color displays only. It selects ‘XY’ format dumping instead of the default ‘Z’ format.</p> <p>-add <i>value</i> This option specifies an signed value to be added to every pixel.</p> <p>-frame This option indicates that the window manager frame should be included when manually selecting a window.</p> <p>-root This option indicates that the root window should be selected for the window dump, without requiring the user to select a window with the pointer.</p> <p>-id <i>id</i> This option indicates that the window with the specified resource id should be selected for the window dump, without requiring the user to select a window with the pointer.</p> <p>-name <i>name</i> This option indicates that the window with the specified WM_NAME property should be selected for the window dump, without requiring the user to select a window with the pointer.</p> <p>-icmap Normally the colormap of the chosen window is used to obtain RGB values. This option forces the first installed colormap of the screen to be used instead.</p> <p>-screen This option indicates that the GetImage request used to obtain the image should be done on the root window, rather than directly on the specified window. In this way, you can obtain pieces of other windows that overlap the specified window, and more importantly, you can capture menus or other popups that are independent windows but appear over the specified window.</p> |

| | |
|--------------------|---|
| ENVIRONMENT | DISPLAY To get default host and display number. |
| CAVEAT | If the dumped window is partially obscured by another window then the obscured area has undefined content. |
| FILES | XWDFile.h X Window Dump File format definition file. |
| SEE ALSO | xwud(1) , xpr(1) , X11(7) |
| COPYRIGHT | Copyright 1988, Massachusetts Institute of Technology. See X11(7) for a full statement of rights and permissions. |
| AUTHORS | Tony Della Fera, Digital Equipment Corp., MIT Project Athena William F. Wyatt, Smithsonian Astrophysical Observatory |

| | |
|--------------------|--|
| NAME | xwininfo – window information utility for X |
| SYNOPSIS | xwininfo [-help] [-id <i>id</i>] [-root] [-name <i>name</i>] [-int] [-children] [-tree] [-stats] [-bits] [-events] [-size] [-wm] [-shape] [-frame] [-all] [-english] [-metric] [-display <i>display</i>] |
| DESCRIPTION | <p><i>Xwininfo</i> is a utility for displaying information about windows. Various information is displayed depending on which options are selected. If no options are chosen, -stats is assumed.</p> <p>The user has the option of selecting the target window with the mouse (by clicking any mouse button in the desired window) or by specifying its window id on the command line with the -id option. Or instead of specifying the window by its id number, the -name option may be used to specify which window is desired by name. There is also a special -root option to quickly obtain information on the screen's root window.</p> |
| OPTIONS | <p>-help Print out the 'Usage:' command syntax summary.</p> <p>-id <i>id</i> This option allows the user to specify a target window <i>id</i> on the command line rather than using the mouse to select the target window. This is very useful in debugging X applications where the target window is not mapped to the screen or where the use of the mouse might be impossible or interfere with the application.</p> <p>-name <i>name</i> This option allows the user to specify that the window named <i>name</i> is the target window on the command line rather than using the mouse to select the target window.</p> <p>-root This option specifies that X's root window is the target window. This is useful in situations where the root window is completely obscured.</p> <p>-int This option specifies that all X window ids should be displayed as integer values. The default is to display them as hexadecimal values.</p> <p>-children This option causes the root, parent, and children windows' ids and names of the selected window to be displayed.</p> <p>-tree This option is like -children but displays all children recursively.</p> <p>-stats This option causes the display of various attributes pertaining to the location and appearance of the selected window. Information displayed includes the location of the window, its width and height, its depth, border width, class, colormap id if any, map state, backing-store hint, and location of the corners.</p> <p>-bits This option causes the display of various attributes pertaining to the selected window's raw bits and how the selected window is to be stored. Displayed information includes the selected window's bit gravity, window gravity, backing-store hint, backing-planes value, backing pixel, and whether or not the window has save-under set.</p> |

- events** This option causes the selected window's event masks to be displayed. Both the event mask of events wanted by some client and the event mask of events not to propagate are displayed.
- size** This option causes the selected window's sizing hints to be displayed. Displayed information includes: for both the normal size hints and the zoom size hints, the user supplied location if any; the program supplied location if any; the user supplied size if any; the program supplied size if any; the minimum size if any; the maximum size if any; the resize increments if any; and the minimum and maximum aspect ratios if any.
- wm** This option causes the selected window's window manager hints to be displayed. Information displayed may include whether or not the application accepts input, what the window's icon window # and name is, where the window's icon should go, and what the window's initial state should be.
- shape** This option causes the selected window's window and border shape extents to be displayed.
- frame** This option causes window manager frames to be considered when manually selecting windows.
- metric** This option causes all individual height, width, and x and y positions to be displayed in millimeters as well as number of pixels, based on what the server thinks the resolution is. Geometry specifications that are in +x+y form are not changed.
- english** This option causes all individual height, width, and x and y positions to be displayed in inches (and feet, yards, and miles if necessary) as well as number of pixels. **-metric** and **-english** may both be enabled at the same time.
- all** This option is a quick way to ask for all information possible.
- display** *display*
This option allows you to specify the server to connect to; see *X(1)*.

EXAMPLE

The following is a sample summary taken with no options specified:

```
xwininfo: Window id: 0x60000f "xterm"
```

```

Absolute upper-left X: 2
Absolute upper-left Y: 85
Relative upper-left X: 0
Relative upper-left Y: 25
Width: 579
Height: 316
Depth: 8
Visual Class: PseudoColor
Border width: 0
Class: InputOutput
```

Colormap: 0x27 (installed)
Bit Gravity State: NorthWestGravity
Window Gravity State: NorthWestGravity
Backing Store State: NotUseful
Save Under State: no
Map State: IsViewable
Override Redirect State: no
Corners: +2+85 -699+85 -699-623 +2-623
-geometry 80x24+0+58

ENVIRONMENT**DISPLAY**

To get the default host and display number.

SEE ALSO

X11(7), **xprop(1)**

BUGS

Using `-stats -bits` shows some redundant information.

The `-geometry` string displayed must make assumptions about the window's border width and the behavior of the application and the window manager. As a result, the location given is not always correct.

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
See *X11(7)* for a full statement of rights and permissions.

AUTHOR

Mark Lillibridge, MIT Project Athena

| | |
|--------------------|--|
| NAME | xwud – image displayer for X |
| SYNOPSIS | xwud [-in <i>file</i>] [-noclick] [-geometry <i>geom</i>] [-display <i>display</i>] [-new] [-std <i>maptype</i>] [-raw] [-vis <i>vis-type-or-id</i>] [-help] [-rv] [-plane <i>number</i>] [-fg <i>color</i>] [-bg <i>color</i>] |
| DESCRIPTION | Xwud is an X Window System image undumping utility. Xwud allows X users to display in a window an image saved in a specially formatted dump file, such as produced by xwd (1). |
| OPTIONS | <p>-bg <i>color</i> If a bitmap image (or a single plane of an image) is displayed, this option can be used to specify the color to display for the "0" bits in the image.</p> <p>-display <i>display</i> This option allows you to specify the server to connect to; see X11(1).</p> <p>-fg <i>color</i> If a bitmap image (or a single plane of an image) is displayed, this option can be used to specify the color to display for the "1" bits in the image.</p> <p>-geometry <i>geom</i> This option allows you to specify the size and position of the window. Typically you will only want to specify the position, and let the size default to the actual size of the image.</p> <p>-help Print out a short description of the allowable options.</p> <p>-in <i>file</i> This option allows the user to explicitly specify the input file on the command line. If no input file is given, the standard input is assumed.</p> <p>-new This option forces creation of a new colormap for displaying the image. If the image characteristics happen to match those of the display, this can get the image on the screen faster, but at the cost of using a new colormap (which on most displays will cause other windows to go technicolor).</p> <p>-noclick Clicking any button in the window will terminate the application, unless this option is specified. Termination can always be achieved by typing 'q', 'Q', or ctrl-c.</p> <p>-plane <i>number</i> You can select a single bit plane of the image to display with this option. Planes are numbered with zero being the least significant bit. This option can be used to figure out which plane to pass to xpr(1) for printing.</p> <p>-raw This option forces the image to be displayed with whatever color values happen to currently exist on the screen. This option is mostly useful when undumping an image back onto the same screen that the image originally came from, while the original windows are still on the screen, and results in getting the image on the screen faster.</p> <p>-rv If a bitmap image (or a single plane of an image) is displayed, this option forces the foreground and background colors to be swapped. This may be needed when displaying a bitmap image which has the color sense of pixel values "0"</p> |

and "1" reversed from what they are on your display.

-std *maptype*

This option causes the image to be displayed using the specified Standard Colormap. The property name is obtained by converting the type to upper case, prepending "RGB_", and appending "_MAP". Typical types are "best", "default", and "gray". See **xstdcmap(1)** for one way of creating Standard Colormaps.

-vis *vis-type-or-id*

This option allows you to specify a particular visual or visual class. The default is to pick the "best" one. A particular class can be specified: "StaticGray", "GrayScale", "StaticColor", "PseudoColor", "DirectColor", or "TrueColor". Or "Match" can be specified, meaning use the same class as the source image. Alternatively, an exact visual id (specific to the server) can be specified, either as a hexadecimal number (prefixed with "0x") or as a decimal number. Finally, "default" can be specified, meaning to use the same class as the colormap of the root window. Case is not significant in any of these strings.

ENVIRONMENT

DISPLAY

To get default display.

FILES

XWDFFile.h

X Window Dump File format definition file.

SEE ALSO

xwd(1), **xpr(1)**, **xstdcmap(1)**, **X11(7)**

COPYRIGHT

Copyright 1988, Massachusetts Institute of Technology.
See **X11(7)** for a full statement of rights and permissions.

AUTHOR

Bob Scheifler, MIT X Consortium

Index

Special Characters

24to8(1) — converts Sun rasterfiles to 8-bit format, 8

A

ab_admin(1) — administer AnswerBook card catalog database(s), 9

ab_cardcatalog(4) — file of available AnswerBooks, 11

ab_library(4) — library of AnswerBooks searched by **navigator** — **ab_library(4)**, 13

accessx(1) — interface to keyboard enhancements, 14

align_equals — **textedit** selection filter, 296

answerbook(1) — on-line documentation viewing system, 16

AnswerBook

administer card catalog database(s) —

ab_admin(1), 9

browse and search AnswerBook documentation — **navigator(1)**, 175

documentation viewer — **docviewer(1)**, 91

file of available AnswerBooks —

ab_cardcatalog(4), 11

general information — **answerbook(1)**

library of AnswerBooks searched by **navigator** — **ab_library(4)**, 13

print AnswerBook documents using **lp** — **viewprint(1)**, 342

appres(1) — application resource database utility, 18

atobm(1) — ASCII to bitmap conversion utility, 27

audiocontrol(1) — audio control tool, 19

audiotool(1) — audio recording & playback tool, 22

B

bdfpcf(1) — font conversion utility, 24

bdfosnf(1) — font conversion utility, 25

binder(1) — modify deskset bindings database, 26

bitmap(1) — bitmap editor & conversion utilities, 27

bldrgb(1) — colorname database utility, 38

bmtoa(1) — bitmap to ASCII conversion utility, 27

C

calctool(1) — desktop calculator tool

Calendar Manager

browse calendar appointments —

cm_lookup(1), 81

daemon used with **cm** — **rpc.cmsd(1)**, 274

delete calendar appointments — **cm_delete(1)**, 74

general information — **cm(1)**

insert calendar appointments — **cm_insert(1)**, 78

capitalize — **textedit** selection filter, 296

ce_db_build(1) — build CE database from ASCII files, 50

ce_db_merge(1) — merge new data into CE database, 54

clock(1) — desktop clock program, 58

cm(1) — desktop calendar tool, 62

cm_delete(1) — delete calendar appointments from command line, 74

cm_insert(1) — inserts calendar appointments from command line, 78

cm_lookup(1) — browse calendar appointments from command line, 81

cmap_alloc(1) — colormap utility, 67

cmap_compact(1) — utility to reduce colormap flashing, 68

cmdtool(1) — enhanced terminal window program, 70

CoEd(6) — ToolTalk-based text editor, 84

color

- building the colorname database — **rgb**(1), 273
- characterization data — **xcmsdb**(1), 382
- colorname database utility — **bldrgb**(1), 38
- converting Sun rasterfiles to 8-bit format — **24to8**(1), 8
- default colormap allocation — **cmap_alloc**(1), 67
- displaying the colorname database — **showrgb**(1), 280
- display current colors — **xcolor**(1), 384
- icon color manipulation tool — **colorchooser**(1), 85
- preventing flashing — **cmap_compact**(1), 68
- standard colormap properties — **xstdcmap**(1), 522
- swap red & blue — **redxblue**(1), 270
- utility to reduce colormap flashing — **ds_server_init**(1), 100

colorchooser(1) — icon color manipulation tool, 85

constype(1) — prints console type, 86

convert_to_Xdefaults(1) — SunView settings converter, 87

convert_to_xview(1) — SunView to Xview converter, 89

D

demonstration programs

- client/server demo programs — **edit_demo**(6), 109
- command-line interface for PostScript — **dpsexec**(6), 98
- draws plaid patterns — **plaid**(6), 248
- fishtank animation demo — **realxfishdb**(6), 268
- graphics primitives demo — **xgc**(6), 425
- improved polyhedron demo — **ico2**(6), 128
- MacPaint display demo — **xmac**(6), 446
- maze demo — **maze**(6), 165
- monitor ToolTalk messages — **ttsnoop**(6), 312
- OLIT demo program — **olitsampler**(6), 182
- pattern drawing demo — **muncher**(6), 174
- puzzle demo — **puzzle**(1), 262
- rotating polyhedron — **ico**(6), 127
- screen magnification demo — **xmag_multivis**(6), 449
- simple ToolTalk demo — **ttsample**(6), 308
- solitaire game demo — **xsol**(6), 521
- spinning text game — **texteroids**(6), 298
- text editor demo — **CoEd**(6), 84
- worm drawing demo — **worms**(6), 350

docviewer(1) — viewer for AnswerBook documentation, 91

dps(7) — general information about Display Postscript, 93

dpsexec(6) — command-line interface for PostScript, 98

ds_server_init(1) — utility to reduce colormap flashing, 100

dsdm(1) — drop site database manager, 99

E

edit_demo(6) — ToolTalk client/server demo programs, 109

editres(1) — X resource editor, 101

F

filemgr(1) — file management tool, 110

fixframe — converts FrameMaker *.ps files to PostScript, 120

fmgc(1) — clean up files left by **filemgr** — **fmgc(1)**, 122

fonts

bitmap font files — **makebdf(1)**, 157
converting to PCF — **bdfpcf(1)**, 24
converting to SNF — **bdfosnf(1)**, 25
display utility — **xfd(1)**, 420
fonts.dir file — **mkfontdir(1)**, 168
property display utility — **xprop(1)**, 496
selection utility — **xfontsel(1)**, 422
server font listing utility — **xlsfonts(1)**, 443
show contents of SNF file — **showsnf(1)**, 281

format_floppy(1) — diskette formatting program, 123

H

helpopen(1) — program to communicate with help-viewer, 124

helpviewer(1) — OpenWindows viewer for on-line help, 125

I

ico2(6) — improved polyhedron animation demo, 128

ico(6) — rotating polyhedron demo program, 127

iconedit(1) — DeskSet utility for editing icons, 130

images

capturing screens — **snapshot(1)**, 282

OpenWindows viewing program —

imagetool(1), 132

rasterfile image format — **rasterfile(4)**, 267

rasterfile conversion utility — **rash(1)**, 265

imagetool(1) — Image viewing application, 132

imake(1) — **make** utility interface, 136

insert_brackets — **textedit** selection filter, 296

K

kbd_mode(1) — keyboard translation utility, 139

keyboard

keytable mapping file — **keytable.map(4)**, 140

resetting input — **kbd_mode(1)**, 139

translation utility — **kbd_mode(1)**, 139

virtual keys & function displayer — **vkbd(1)**,

347

keyboard, *continued*

keytable.map(4) — keytable mapping file, 140

L

listres(1) — lists widget resources, 142

locale_env(1) — sets language, 143

M

mail

mailbox monitoring utility — **xbiff(1)**, 366

OLIT mailbox monitor — **olbiff(6)**, 181

OPEN LOOK interface for MH — **olmh(6)**

OpenWindows mail interface — **mailtool(1)**, 147

removing attachments — **mailprint(1)**, 146

X System mail utility — **xmh(1)**

mailprint(1) — mail message attachment filter, 146

mailtool(1) — OpenWindows mail interface, 147

makebdf(1) — creates bitmap files from scalable fonts, 157

makedepend(1) — create makefile dependencies, 159

maze(6) — automated maze demo, 165

mkfontdir(1) — create fonts.dir file, 168

mp(1) — PostScript printer program, 170

muncher(6) — pattern drawing demo, 174

N

navigator(1) — browse and search AnswerBook documentation, 175

O

oclock(1) — analog clock, 179

olbiff(6) — OLIT mailbox monitor utility, 181

olitsampler(6) — OLIT widget demo, 182

olittable(6) — periodic table demo, 183

olmh(6) — OPEN LOOK interface for MH system

olwm(1) — OPEN LOOK window manager

olwmslave(1) — **olwm** utility program, 229

openwin(1) — OpenWindows startup script, 230

OpenWindows

OpenWindows, *continued*

- audio control tool — **audiocontrol**(1), 19
 - calculator tool — **calctool**(1), 39
 - client listing utility — **owplaces**(1), 234
 - clock program — **clock**(1), 58
 - file management tool — **filemgr**(1), 110
 - mail interface — **mailtool**(1), 147
 - on-line help viewer — **helpviewer**(1), 125
 - OPEN LOOK window manager — **olwm**(1)
 - printing tool — **printtool**(1), 249
 - property setting utility — **props**(1), 252
 - screen capturing program — **snapshot**(1), 282
 - shell window program — **shelltool**(1), 277
 - startup script — **openwin**(1), 230
 - tape drive control tool — **tapetool**(1), 287
- owplaces**(1) — X client listing utility, 234

P

- pageview**(1) — PostScript language previewer, 236
- perfmeter**(1) — system performance display program, 240
- plaid**(6) — draws plaid-like patterns, 248
- PostScript
- command-line interface — **dpsexec**(6), 98
 - convert FrameMaker *.ps files — **fixframe**(1), 120
 - converting rasterfiles — **rash**(1), 265
 - create C procedures — **pswrap**(1), 260
 - encapsulated PostScript display program — **xepsf**(6), 417
 - general information about DPS — **dps**(7), 93
 - printing program — **mp**(1), 170
 - rasterfile conversion — **ras2ps**(1), 263
- printtool**(1) — OpenWindows printing tool, 249
- props**(1) — OpenWindows property setting utility, 252
- pswrap**(1) — creates C procedures from PostScript code, 260
- puzzle**(6) — puzzle game demo, 262

R

- ras2ps**(1) — convert rasterfile to PostScript, 263
- rash**(1) — rasterfile to PostScript translator, 265
- rasterfile, 267

- realxfishdb**(6) — animated fishtank display, 268
- redxblue**(1) — swaps red & blue in rasterfiles, 270
- remove_brackets** — **textedit** selection filter, 296
- resize**(1) — resets terminal settings to window size, 271
- rgb**(1) — build colorname database, 273
- rpc.cmsd**(1) — calendar manager service daemon, 274

S

- sessreg**(1) — manage utmp/wtmp entries, 275
- shelltool**(1) — OpenWindows shell program, 277
- shift_lines** — **textedit** selection filter, 296
- showrgb**(1) — display the colorname database, 280
- showsuf**(1) — show contents of SNF file, 281
- snapshot**(1) — screen image capture tool, 282
- spider**(6) — double-deck solitaire card game, 285

T

- tapetool**(1) — tape drive control tool, 287
- text
- command window — **cmdtool**(1), 70
 - copy selection to buffer — **xcutsel**(1), 387
 - filters for text processing — **textedit_filters**(1), 296
 - ToolTalk-based editor — **CoEd**(6), 84
 - Xview-based text editor — **textedit**(1), 290
- textedit**(1) — Xview-based text editor, 290
- textedit_filters**(1) — text processing filters, 296
- texteroids**(6) — spinning text game, 298
- ToolTalk
- archive files using ToolTalk — **tttar**(1), 314
 - check or repair ToolTalk databases — **ttdbck**(1M), 303
 - client/server demo programs — **edit_demo**(6), 109
 - compiler — **tt_type_comp**(1), 316
 - converts CE type tables to XDR format — **ttce2xdr**(1M), 300
 - demo to monitor messages — **ttsnoop**(6), 312
 - message server — **ttsession**(1), 309
 - move or rename files — **ttmv**(1), 306
 - remove files using ToolTalk — **ttm**(1), 307

ToolTalk, *continued*

RPC-based ToolTalk database server —
ttdbserverd(1M), 305
simple ToolTalk demo — **ttsample**(6), 308
text editor demo — **CoEd**(6), 84
toolwait(1) — X client startup utility, 299
tt_type_comp(1) — ToolTalk otype & ptype compiler, 316
ttce2xdr(1M) — convert CE type tables to XDR format, 300
ttdbck(1M) — check or repair ToolTalk databases, 303
ttdbserverd(1M) — RPC-based ToolTalk database server, 305
ttmv(1) — move or rename files using ToolTalk, 306
trrm(1) — remove files using ToolTalk, 307
trmdir(1) — remove directories using ToolTalk, 307
ttsample(6) — simple ToolTalk demo, 308
ttsession(1) — ToolTalk message server, 309
ttnoop(6) — ToolTalk message monitor demo, 312
tttar(1) — archive files using ToolTalk, 314
twm(1) — tab window manager, 319

V

viewprint(1) — print AnswerBook documents using lp, 342
viewprint command, 342
viewres(1) — display widget class hierarchy, 343
vkbd(1) — virtual keys & functions displayer, 347

W

windows
changing fonts or colors — **props**(1), 252
check system protocols — **winsysck**(1), 348
dump image of X window — **xwd**(1), 572, 409
information display utility — **xwininfo**(1), 574
listing utility — **xlswins**(1), 445
OPEN LOOK window manager — **olwm**(1)
print X window dump — **xpr**(1), 491
property display utility — **xprop**(1), 496
protocol display demo — **xscope**(6), 507
root window settings — **xsetroot**(1), 519
set terminal settings to window size —

resize(1),
windows, *continued*
271
shell window program — **shelltool**(1), 277
tab window manager — **twm**(1), 319
terminal window program — **cmdtool**(1), 70
X Window System — **X11**(7), 351
winsysck(1) — check window protocols, 348
worm(6) — worm drawing demo, 350

X

X11(7) — general information about X
xauth(1) — Xserver authorization utility, 363
xbiff(1) — mailbox monitoring utility, 366
xcalc(91) — desktop calculator tool, 2
xclipboard(1) — clipboard display utility, 377
xclock(1) — desktop clock, 379
xcmsdb(1) — manipulate color characterization, 382
xcolor(1) — display current colors, 384
xcolsole(1) — display console messages, 385
xcutsel(1) — copy selection to buffer, 387
xditview(1) — display ditroff files, 389
xdm(1) — X display manager, 392
xdpr(1) — X window dump-to-printer utility, 409
xdpyinfo(1) — Xserver information utility, 410
xedit(1) — simple text editor, 414
xepsf(6) — encapsulated PostScript display demo, 417
xev(6) — prints contents of X events, 418
xfd(1) — font display utility, 420
xfontsel(1) — font selection utility, 422
xgc(6) — graphics primitives demo, 425
xhost(1) — controls hosts connected to the Xserver, 427
xinit(1) — Xserver initialization program, 429
xkill(1) — forces Xserver to close connections, 432
xload(1) — display system load average, 433
xlock(1) — screen locking utility, 435
xlogo(1) — display X Sysyem logo, 439
xlsatoms(1) — list interned atoms on server, 441
xlsclients(1) — list client application, 442
xlsfonts(1) — server font list displayer, 443

xlswins(1) — X window listing utility, 445
xmac(6) — MacPaint file display demo, 446
xmag(1) — screen magnification utility, 447
xmag_multivis(6) — screen magnification demo, 449
xman(1) — display manpages for X, 453
xmh(1) — X Systems mail utility
xmkmf(1) — interface to **imake** utility, 486
xpr(1) — prints X window dump, 491
xprop(1) — display window & font properties, 496
xrdb(1) — Xserver resource database utility, 501
xscope(6) — X window protocol display demo, 507
Xserver
 authorization display utility — **xauth**(1), 363
 client listing application — **xlsclients**(1), 442
 closing connections — **xkill**(1), 432
 display manager — **xdm**(1), 392
 event printing demo — **xev**(6), 418
 general information — **Xserver**(1), 509
 hosts allowed to make connections — **xhost**(1), 427
 information display utility — **xcpyinfo**(1), 410
 initialization program — **xinit**(1), 429
 listing interned atoms — **xlsatoms**(1), 441
 property display utility — **xprop**(1), 496
 protocol display program — **xscope**(6), 507
 database utility — **xrdb**(1), 501
 Solaris implementation — **Xsun**(1), 524
xset(1) — user preference utility, 516
xsetroot(1) — sets parameters of root's window, 519
xsol(6) — solitaire card game, 521
xstdcmap(1) — define standard colormap properties, 522
Xsun(1) — Solaris Xserver, 524
xterm(1) — terminal emulator for X, 526
xv_get_sel(1) — copy selection to standard output, 571
xview(7) — xview toolkit information, 551
xwd(1) — dump image of X window, 572
xwininfo(1) — X window information utility, 574
xwud(1) — image display utility, 577