Preface

OVERVIEW

A man page is provided for both the naive user, and sophisticated user who is familiar with the OpenWindows Desktop system and is in need of on-line information. A man page is intended to answer concisely the question “What does it do?” The man pages in general comprise a reference manual. They are not intended to be a tutorial.

The following contains a brief description of each section in the man pages and the information it references:

- Section 1 describes, in alphabetical order, commands available with the openwindows desktop system.
- Section 1M describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.
- Section 4 outlines the formats of various files.
- Section 6 describes various games and demos.
- Section 7 describes various special files that refer to specific hardware peripherals, and device drivers.

Below is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if there are no bugs to report, there is no BUGS section. See the intro...
pages for more information and detail about each section, and man(1) for more information about man pages in general.

NAME

This section gives the names of the commands or functions documented, followed by a brief description of what they do.

SYNOPSIS

This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full pathname is shown. Literal characters (commands and options) are in **bold** font and variables (arguments, parameters and substitution characters) are in *italic* font. Options and arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required.

The following special characters are used in this section:

- **[]** The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.
- **...** Ellipses. Several values may be provided for the previous argument, or the previous argument can be specified multiple times, for example, ‘filename ...’.
- **|** Separator. Only one of the arguments separated by this character can be specified at time.
- **{}** Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.

AVAILABILITY

This section briefly states any limitations on the availability of the command. These limitations could be hardware or software specific.

A specification of a class of hardware platform, such as **x86** or **SPARC**, denotes that the command or interface is applicable for the hardware platform specified.

In Section 1 and Section 1M, **AVAILABILITY** indicates which package contains the command being described on the manual page. In order to use the command, the specified package must have been installed with the operating system. If the package was not installed, see **pkgadd**(1) for information on how
to upgrade.

**DESCRIPTION**

This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, functions and such, are described under USAGE.

**OPTIONS**

This lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.

**RETURN VALUES**

If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or −1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared as `void` do not return values, so they are not discussed in RETURN VALUES.

**ERR0RS**

On failure, most functions place an error code in the global variable `errno` indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.
 USAGE

This section is provided as a guidance on use. This section lists special rules, features and commands that require in-depth explanations. The subsections listed below are used to explain built-in functionality:

- Commands
- Modifiers
- Variables
- Expressions
- Input Grammar

 EXAMPLES

This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command line entry and machine response is shown. Whenever an example is given, the prompt is shown as

```
example%
```

or if the user must be super-user,

```
example#
```

Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS and USAGE sections.

 ENVIRONMENT

This section lists any environment variables that the command or function affects, followed by a brief description of the effect.

 FILES

This section lists all filenames referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.

 SEE ALSO

This section lists references to other man pages, in-house documentation and outside publications.
DIAGNOSTICS

This section lists diagnostic messages with a brief explanation of the condition causing the error. Messages appear in **bold** font with the exception of variables, which are in *italic* font.

WARNINGS

This section lists warnings about special conditions which could seriously affect your working conditions — this is not a list of diagnostics.

NOTES

This section lists additional information that does not belong anywhere else on the page. It takes the form of an aside to the user, covering points of special interest. Critical information is never covered here.

BUGS

This section describes known bugs and wherever possible suggests workarounds.
NAME
Intro – introduction to the OpenWindows Desktop Reference Manual

OVERVIEW
The Solaris OpenWindows system includes some sophisticated clients referred to as the Solaris DeskSet suite. These are workgroup productivity tools like Calendar Manager and Mailtool that take advantage of the value-added enhancements to the Solaris X server.

This manual and the Solaris X Window System Reference Manual cover all man pages that make up the SUNWaman software package, other than those in Section 3. This package is usually installed as part of the "Developer" software set during initial installation of Solaris. The default installation directory for these man pages is /usr/openwin/man. To access these man pages, add /usr/openwin/man to the SMANPATH environment variable.

<table>
<thead>
<tr>
<th>Name</th>
<th>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>answerbook(1)</td>
<td>browse, search, and view on-line documentation</td>
</tr>
<tr>
<td>audiocontrol(1)</td>
<td>audio control panel</td>
</tr>
<tr>
<td>audiotool(1)</td>
<td>audio play/record tool</td>
</tr>
<tr>
<td>binder(1)</td>
<td>modify deskset bindings database</td>
</tr>
<tr>
<td>calctool(1)</td>
<td>a desktop calculator</td>
</tr>
<tr>
<td>ce_db_build(1)</td>
<td>build a Classing Engine database from an ASCII description of the database or generate an ASCII description file of a CE database</td>
</tr>
<tr>
<td>ce_db_merge(1)</td>
<td>merge new information into a Classing Engine database from an ASCII description of the new information</td>
</tr>
<tr>
<td>clock(1)</td>
<td>display the time in an icon or window</td>
</tr>
<tr>
<td>cm(1)</td>
<td>calendar manager, appointment and resource scheduling tool</td>
</tr>
<tr>
<td>cmdtool(1)</td>
<td>run a shell (or other program) in an OpenWindows enhanced terminal window</td>
</tr>
<tr>
<td>cm_delete(1)</td>
<td>delete appointments from Calendar Manager database</td>
</tr>
<tr>
<td>cm_insert(1)</td>
<td>insert appointments into Calendar Manager database</td>
</tr>
<tr>
<td>cm_lookup(1)</td>
<td>look up appointments from Calendar Manager database</td>
</tr>
<tr>
<td>colorChooser(1)</td>
<td>change icon colors in deskset tools</td>
</tr>
<tr>
<td>convert_to_Xdefaults(1)</td>
<td>convert SunView1 defaults into equivalent Xdefaults</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>convert_to_xview</td>
<td>convert a SunView1 source program to XView source</td>
</tr>
<tr>
<td>digestp</td>
<td>See mailp(1)</td>
</tr>
<tr>
<td>docviewer</td>
<td>OpenWindows viewer for AnswerBook on-line documentation</td>
</tr>
<tr>
<td>dsdm</td>
<td>Drop Site Database Manager</td>
</tr>
<tr>
<td>ds_server_init</td>
<td>store a property in the X11 server to reduce colormap flashing within the DeskSet applications</td>
</tr>
<tr>
<td>edit_demo</td>
<td>ToolTalk client/server demonstration programs</td>
</tr>
<tr>
<td>filemgr</td>
<td>a GUI-based file management application</td>
</tr>
<tr>
<td>filep</td>
<td>See mailp(1)</td>
</tr>
<tr>
<td>filofaxp</td>
<td>See mailp(1)</td>
</tr>
<tr>
<td>fish_props</td>
<td>See realxfishdb(6)</td>
</tr>
<tr>
<td>fixframe</td>
<td>convert FrameMaker 2.0 PostScript files to conform to the PostScript structuring conventions</td>
</tr>
<tr>
<td>fixinterleaf</td>
<td>convert Interleaf PostScript files to conform to the PostScript structuring conventions</td>
</tr>
<tr>
<td>fmgc</td>
<td>a program to garbage collect a users <code>/.fm</code> directory hierarchy</td>
</tr>
<tr>
<td>format_floppy</td>
<td>diskette formatting program used by FileManager</td>
</tr>
<tr>
<td>franklinp</td>
<td>See mailp(1)</td>
</tr>
<tr>
<td>helpopen</td>
<td>utility for programmatically controlling 3helpviewerP</td>
</tr>
<tr>
<td>helpviewer</td>
<td>OpenWindows viewer for on-line help handbooks</td>
</tr>
<tr>
<td>iconedit</td>
<td>create and edit images for OpenWindows icons, cursors, and panel items</td>
</tr>
<tr>
<td>imagetool</td>
<td>Image viewer for OpenWindows</td>
</tr>
<tr>
<td>locale_env</td>
<td>locale_env - program for openwin-sys</td>
</tr>
<tr>
<td>mailp</td>
<td>frontends to the mp PostScript pretty printer</td>
</tr>
<tr>
<td>mailprint</td>
<td>filter to strip out mail message attachments</td>
</tr>
<tr>
<td>mailtool</td>
<td>OpenWindows interface for the mail program</td>
</tr>
<tr>
<td>mp</td>
<td>PostScript pretty printer</td>
</tr>
<tr>
<td>navigator</td>
<td>browse and search AnswerBook on-line documentation</td>
</tr>
<tr>
<td>newsp</td>
<td>See mailp(1)</td>
</tr>
<tr>
<td>olbiff</td>
<td>OLIT mailbox monitor</td>
</tr>
<tr>
<td>olitsampler</td>
<td>OLIT widget demo program</td>
</tr>
<tr>
<td>olittable</td>
<td>OLIT widget demo program</td>
</tr>
<tr>
<td>olmh</td>
<td>OpenLook interface to the MH message handling system</td>
</tr>
<tr>
<td>olwm</td>
<td>OPEN LOOK window manager for OpenWindows</td>
</tr>
<tr>
<td>olwmslave</td>
<td>olwmslave - helper program for olwm</td>
</tr>
</tbody>
</table>

modified 28 August 1995
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>openwin(1)</td>
<td>OpenWindows startup script</td>
</tr>
<tr>
<td>owplaces(1)</td>
<td>list client applications running on a display</td>
</tr>
<tr>
<td>pageview(1)</td>
<td>PostScript language previewer for OpenWindows</td>
</tr>
<tr>
<td>perfmeter(1)</td>
<td>display system performance values in multiple dials or strip charts</td>
</tr>
<tr>
<td>printtool(1)</td>
<td>OpenWindows tool for printing files</td>
</tr>
<tr>
<td>props(1)</td>
<td>graphical interface to set OpenWindows properties</td>
</tr>
<tr>
<td>rash(1)</td>
<td>Sun Raster to PostScript translator</td>
</tr>
<tr>
<td>realxfishdb(6)</td>
<td>Display a fishtank on the root window of an X11 server.</td>
</tr>
<tr>
<td>rpc.cmsd(1)</td>
<td>calendar manager service daemon</td>
</tr>
<tr>
<td>rpc.ttdbserverd(1M)</td>
<td>See ttdbserverd(1M)</td>
</tr>
<tr>
<td>shelltool(1)</td>
<td>run a shell (or other program) in an OpenWindows terminal window</td>
</tr>
<tr>
<td>snapshot(1)</td>
<td>capture some or all of a screen image and save to a raster file</td>
</tr>
<tr>
<td>spider(6)</td>
<td>play double deck solitaire</td>
</tr>
<tr>
<td>tapetool(1)</td>
<td>OpenWindows tool for reading files from tape or archiving files to tape</td>
</tr>
<tr>
<td>textedit(1)</td>
<td>XView-based text editor with mouse support</td>
</tr>
<tr>
<td>textedit_filters(1)</td>
<td>filters provided with textedit(1)</td>
</tr>
<tr>
<td>timemanp(1)</td>
<td>See mailp(1)</td>
</tr>
<tr>
<td>timesysp(1)</td>
<td>See mailp(1)</td>
</tr>
<tr>
<td>toolwait(1)</td>
<td>control client program startup</td>
</tr>
<tr>
<td>tce2xdr(1M)</td>
<td>convert ToolTalk Classing Engine type tables to XDR format</td>
</tr>
<tr>
<td>ttcp(1)</td>
<td>copy files and inform the ToolTalk service</td>
</tr>
<tr>
<td>ttdbck(1M)</td>
<td>display, check, or repair ToolTalk databases</td>
</tr>
<tr>
<td>ttdbserverd(1M)</td>
<td>RPC-based ToolTalk database server</td>
</tr>
<tr>
<td>ttmv(1)</td>
<td>move or rename files and inform the ToolTalk service</td>
</tr>
<tr>
<td>ttrm(1)</td>
<td>remove files or directories and inform the ToolTalk service</td>
</tr>
<tr>
<td>ttrmdir(1)</td>
<td>remove empty directories and inform the ToolTalk service</td>
</tr>
<tr>
<td>ttsample1(6)</td>
<td>simple ToolTalk demo</td>
</tr>
<tr>
<td>ttsession(1)</td>
<td>the ToolTalk message server</td>
</tr>
<tr>
<td>tt.tar(1)</td>
<td>process files and ToolTalk objects in an archive</td>
</tr>
<tr>
<td>tttrace(1)</td>
<td>trace ToolTalk calls and messages</td>
</tr>
<tr>
<td>tt_type_comp(1)</td>
<td>compile ToolTalk otypes and ptypes</td>
</tr>
<tr>
<td>viewprint(1)</td>
<td>print AnswerBook documents using lp</td>
</tr>
<tr>
<td>vkbd(1)</td>
<td>virtual keyboard/soft function keys display manager</td>
</tr>
<tr>
<td>xview(7)</td>
<td>xview toolkit information</td>
</tr>
</tbody>
</table>

modified 28 August 1995
**xv_get_sel(1)**  
copy the content of a selection to the standard output
NAME
24to8 – convert a 24- or 32-bit rasterfile to an 8 bit rasterfile suitable for X11

SYNOPSIS
24to8 [ −v ] [ −q ] [ −large ] [ inrasf ] [ − ] [ outrasf ]

DESCRIPTION
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).

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.

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.

OPTIONS
−v Verbose mode will print information as it processes the image. (The default is to be silent.)
−q Query (prints list of options)
−large Dithers into a static colormap with a large color cube of the form 5x9x5.

SEE ALSO
redxblue(1)
NAME  
ab_admin – administer AnswerBook card catalog database(s)

SYNOPSIS  
ab_admin  
  -file card-catalog-file  
     -match answerbook-id[,version]  
     -add answerbook-id[,version]  
       answerbook-attribute-list  
     -modify answerbook-id[,version]  
       answerbook-attribute-list  
     -merge card-catalog-file  
     -convert bookinfo-file  
     -remove answerbook-id[,version]  
     -verify answerbook-id[,version]  
     -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 card-catalog-file  
  Specifies the name of the card catalog file. All ab_admin operations except "-listpaths" require that the file be specified.

-merge card-catalog-file  
  Merge the entries from input card catalog file into the card catalog file being updated.

-convert bookinfo-file  
  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 answerbook-id[,version] [answerbook-attribute-list]  
  Construct a new card catalog entry for the AnswerBook using the information provided on the command line.

-remove answerbook-id[,version]  
  Delete the specified entry.

-match answerbook-id[,version]  
  Display the contents of the specified entry.

-modify answerbook-id[,version] [answerbook-attribute-list]  
  Modify the specified fields of the specified entry.

-verify answerbook-id[,version]  
  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=loc-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)**
<table>
<thead>
<tr>
<th>NAME</th>
<th>ab_cardcatalog – directory of available AnswerBooks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>$HOME/.ab_cardcatalog, $AB_CARDCATALOG</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>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. 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. 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. 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: $HOME/.ab_cardcatalog entries for personal AnswerBooks $AB_CARDCATALOG shared, network-wide AnswerBooks 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 Software and AnswerBook Packages Administration Guide for more information on sharing AnswerBooks on the network.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Each AnswerBook entry in a card catalog file contains the following fields: title AnswerBook title id AnswerBook id version AnswerBook version number (optional) pspath Full path name of directory containing AnswerBook’s PostScript files toopath Full path name of directory containing AnswerBook’s Table of Contents databases indexpath Full path name of directory containing search index files The &quot;version&quot; field is optional and serves mainly to distinguish two AnswerBooks that have the same id but different content. All other fields are mandatory. The format of an entry is a set of colon-separated &quot;field name=value&quot; 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.</td>
</tr>
</tbody>
</table>

modified 29 March 1993
The first line of a card catalog file must be the string

```bash
#<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:

```bash
#<Card Catalog> version 1
#
# Sample card catalog entry for My AnswerBook
#
title=Title of My AnswerBook: \nid=MyAB: \npspath=/net/my_server/export/AnswerBooks/MyAB/ps: \ntocpath=/net/my_server/export/AnswerBooks/MyAB/toc: \nindexpath=/net/my_server/export/AnswerBooks/MyAB/index
#
# Sample card catalog entry for Your AnswerBook
#
title=Title of Your AnswerBook: \nid=YourAB: \npspath=/net/your_server/export/AnswerBooks/YourAB/ps: \ntocpath=/net/your_server/export/AnswerBooks/YourAB/toc: \nindexpath=/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.

modified 29 March 1993
NAME     ab_library – collection of AnswerBooks browsed and searched as a unit

SYNOPSIS  $HOME/.ab_library

DESCRIPTION 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.

The default library file is $HOME/.ab_library

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.

The AnswerBook library file is created and maintained solely by the AnswerBook Navigator. It is not intended to be hand-editable.

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.

FILES  $HOME/.ab_library
Default AnswerBook library file.

$HOME/.ab_library.lock
AnswerBook library advisory locking file.

SEE ALSO  navigator(1)

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

modified 22 Mar 1993
NAME  
answerbook – browse, search, and view on-line documentation

SYNOPSIS  
answerbook [ navigator-options ]

DESCRIPTION  
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.

answerbook performs the following functions each time it runs:

Verify OpenWindows environment
Makes sure that $OPENWINHOME is set, that the correct version of OpenWindows (version 3.3 or greater) is running, etc.

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.

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.

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.

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.

Start AnswerBook Navigator
Invoke navigator(1), passing on any command line options with which it was invoked.

OPTIONS  
answerbook passes all command line options on to navigator(1).

ENVIRONMENT  
AB_CARDCATALOG
answerbook makes locally installed AnswerBooks accessible by appending their card catalog file names to this environment variable, which navigator uses in assembling its list of available AnswerBooks. See ab_cardcatalog(4) for more information.

OPENWINHOME
This environment variable must be set appropriately.

modified 18 November 1993
FILES

$HOME/ab_cardcatalog
Personal card catalog file.

$HOME/.ab_library

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.

/tmp/ab_cardcatalog.*
Temporary card catalog file occasionally generated by `answerbook` to facilitate access to locally installed AnswerBooks.

DIAGNOSTICS
If `answerbook` finds no AnswerBooks during the initialization process, it prints a message to the console and exits without starting up the AnswerBook Navigator.

SEE ALSO `ab_admin(1)`, `docviewer(1)`, `navigator(1)`, `ab_cardcatalog(4)`, `ab_library(4)`

NOTES
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.
AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc.
NAME  audiocontrol – audio control panel

SYNOPSIS  audiocontrol [ −d device ] [ generic-tool-arguments ]

DESCRIPTION  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.

Operations performed by the Audio Control control panel affect all audio programs using the specified device; for instance, adjusting the Play Volume 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.

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.

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.

Play Control Panel

Output Port
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 Speaker and Headphone. Some systems also support Line Out. 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.

Play Volume
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.

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

Mute
This button is used to enable and disable audio output muting. Note that muting output has no effect on audio input recording.

Record...
This button brings up the Record Control Panel.
### 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**
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 ] [ generic-tool-arguments ] [ audio-file ... ]

DESCRIPTION

Audio Tool is an OpenWindows DeskSet application for recording, playing, and simple editing of audio data.

Sound Segment Display

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.

A triangular graphic indicator, called the position pointer, 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.

Control Panel

Play/Stop

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.

While data is playing, this button becomes a Stop button. Audio output may be stopped and restarted at will.

Rewind/Fast-Forward

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.

Record/Stop

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.

File Browser Panel

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. 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
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  binder – modify deskset bindings database

SYNOPSIS  binder [ −user  |−system  |−network  ]

AVAILABILITY  This command is available with the OpenWindows environment.

DESCRIPTION  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.

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.

The three Classing Engine databases are located in:
$OPENWINHOME/lib/cetables/cetables,
/etc/cetables/cetables, and
$HOME/.cetables/cetables, respectively.

By doing this, users may customize their working environment without affecting others.

Applications that use the bindings database in OpenWindows are:
filemgr(1)
binder(1)
imagetool(1)
printtool(1)
mailtool(1)

OPTIONS  
- user  Modify your private database bindings. This is the default mode.
- system  Modify the system database bindings. You must be root to start the binder with this option.
- network  Modify the network database bindings. You must have root access on the OpenWindows server workstation.

FILES  
$OPENWINHOME/lib/cetables/cetables
/etc/cetables/cetables
$HOME/.cetables/cetables

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  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.

modified 10 March 1992
NAME    calctool  –  a desktop calculator

SYNOPSIS    calctool { −2 ] [ −3 ] [ -a accuracy ] [ -c ] [ -l ] [ -m ] [ -name app-name ] [ -r ] [ -v ] [ −? ]
              [ −Wn ] [ +Wn ]

DESCRIPTION    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 associ-
                ated with that key. Each key is discussed in more detail below.

                One of the most important things to remember about calctool is that calculations are per-
                formed from left to right, with no arithmetic precedence. If you need arithmetic pre-
                cedence, then you should use parentheses (see below).

                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 nota-
                tion. The base of operation can be changed between binary, octal, decimal and hexade-
                cimal. Numbers are initially displayed in fixed notation to two numeric places, in the
                decimal base.

                You can use the Copy and Paste 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.

                There are ten memory registers. Numbers can be stored or retrieved in these locations,
                and arithmetic can be performed upon register contents.

                The display windows contains the current numerical value plus the current base and tri-
                gonomometric 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.

                On startup, calctool will use several X resources. These are listed in detail in the
                resources section of these manual pages.

                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 Props function key has the same effect.

                From this property sheet, you can modify several properties of the calculator. Each calcu-
                lator 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.

                Context sensitive help is also available. Position the mouse cursor over the item you need
                help with, and press the Help key. A popup will be displayed giving detailed information
                on this facility.

modified 22 November 1993
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.
program. Note that the ? character must be escaped if using csh(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 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.

modified 22 November 1993
<table>
<thead>
<tr>
<th>Resource</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deskset.calctool.rightHanded</td>
<td>Whether the calculator is started with a &quot;right-handed&quot; display style.</td>
</tr>
<tr>
<td>True, False (True)</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.3dLook</td>
<td>If the display is color, whether each calculator button has a pseudo 3D appearance.</td>
</tr>
<tr>
<td>True, False (True)</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.hasTitle</td>
<td>Indicates whether the calctool window has a title line.</td>
</tr>
<tr>
<td>True, False (True)</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.buttonFont</td>
<td>The name of the font used to display all button labels, irrespective of the current size and scale of calctool.</td>
</tr>
<tr>
<td>Font name string</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.modeFont</td>
<td>The name of the font used to display the various mode labels, irrespective of the current size and scale of calctool.</td>
</tr>
<tr>
<td>Font name string</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.memoryFont</td>
<td>The name of the font used to display the memory register values, irrespective of the current size and scale of calctool.</td>
</tr>
<tr>
<td>Font name string</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.displayFont</td>
<td>The name of the font used to show the current display value, irrespective of the current size and scale of calctool.</td>
</tr>
<tr>
<td>Font name string</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.decDigitColor</td>
<td>The color of the buttons containing the decimal digits 0 - 9 and the numeric point on the main panel.</td>
</tr>
<tr>
<td>Color name string or hexadecimal color specification string</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.hexDigitColor</td>
<td>The color of the buttons containing the hexadecimal digits A - F on the main panel.</td>
</tr>
<tr>
<td>Color name string or hexadecimal color specification string</td>
<td></td>
</tr>
<tr>
<td>deskset.calctool.arithOpColor</td>
<td>Color name string or hexadecimal color specification string</td>
</tr>
<tr>
<td>Description</td>
<td>Resource: deskset.calctool.adjustColor</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Description</td>
<td>The color of the arithmetic operator buttons on the main panel.</td>
</tr>
<tr>
<td>Description</td>
<td>deskset.calctool.portionColor</td>
</tr>
<tr>
<td>Description</td>
<td>The color of the Bsp and Clr buttons on the main panel.</td>
</tr>
<tr>
<td>Description</td>
<td>deskset.calctool.functionColor</td>
</tr>
<tr>
<td>Description</td>
<td>The color of various function buttons on the main panel. These are Acc, Con, Exch, Fun, Keys, Mem, Quit, Rcl and Sto.</td>
</tr>
<tr>
<td>Description</td>
<td>deskset.calctool.mainModeColor</td>
</tr>
<tr>
<td>Description</td>
<td>The color of the three mode buttons on the main panel. These are Base, Disp and Mode.</td>
</tr>
<tr>
<td>Description</td>
<td>deskset.calctool.bitLogicalColor</td>
</tr>
<tr>
<td>Description</td>
<td>The color of the bitwise logical operator buttons in the mode panel. These are And, Not, Or, Xnor and Xor.</td>
</tr>
<tr>
<td>Description</td>
<td>deskset.calctool.finColor</td>
</tr>
<tr>
<td>Description</td>
<td>The color of the financial buttons in the mode panel.</td>
</tr>
<tr>
<td>Description</td>
<td>deskset.calctool.trigModeColor</td>
</tr>
<tr>
<td>Description</td>
<td>The color of the trigonometrical mode buttons in the mode panel. These are Hyp, Inv and Trig.</td>
</tr>
<tr>
<td>Description</td>
<td>deskset.calctool.trigColor</td>
</tr>
<tr>
<td>Description</td>
<td>The color of the trigonometrical buttons in the mode panel. These are Cos, Sin and Tan.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.calctool.sciColor</td>
</tr>
</tbody>
</table>
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 'Ctrl' 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 ('x' 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.
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.

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.

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

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.

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

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:

<table>
<thead>
<tr>
<th>Digit</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.621 km/h per hour / miles per hour</td>
</tr>
<tr>
<td>1</td>
<td>1.41421 square root of 2</td>
</tr>
<tr>
<td>2</td>
<td>2.71828 e.</td>
</tr>
<tr>
<td>3</td>
<td>3.14159 pi.</td>
</tr>
<tr>
<td>4</td>
<td>2.54 cms / inch.</td>
</tr>
<tr>
<td>5</td>
<td>57.29578 degrees in a radian.</td>
</tr>
<tr>
<td>6</td>
<td>1048576.0 2 to the power of 20.</td>
</tr>
<tr>
<td>7</td>
<td>0.0353 gms / oz.</td>
</tr>
<tr>
<td>8</td>
<td>0.948 kilojoules / British thermals.</td>
</tr>
<tr>
<td>9</td>
<td>0.0610 cubic cms / cubic inches.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rcl [ R ]</strong></td>
<td>Retrieve memory register value. There are ten memory registers [0-9].</td>
</tr>
<tr>
<td><strong>Sto [ S ]</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Exch [ X ]</strong></td>
<td>Exchange the current display with the contents of a memory register. There are ten memory registers [0-9].</td>
</tr>
</tbody>
</table>

**Other Operations.**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clr [ Delete ]</strong></td>
<td>Clear the calculator display.</td>
</tr>
<tr>
<td><strong>Bsp [ Back Space ]</strong></td>
<td>Remove the rightmost character of the current entry, and recalculate the displayed value.</td>
</tr>
<tr>
<td><strong>( and ) [ ( and ) ]</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Exp [ E ]</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Keys [ k ]</strong></td>
<td>Toggle the labels on the calctool buttons between the mouse and keyboard equivalents.</td>
</tr>
<tr>
<td><strong>Mem [ m ]</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Quit [ q or Q ]</strong></td>
<td>Exit without user verification.</td>
</tr>
</tbody>
</table>

**FINANCIAL MODE**

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ctm [ 't ]</strong></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Ddb [ 'd ]</strong></td>
<td>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).</td>
</tr>
</tbody>
</table>
**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).

\[ \text{LOGICAL MODE} \]
\[ < \]  
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.

\[ \text{SCIENTIFIC MODE} \]
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.
<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^x$</td>
<td>[ ]</td>
<td>Returns 10 raised to the power of the current entry.</td>
</tr>
<tr>
<td>$y^x$</td>
<td>[ y ]</td>
<td>Take the last entry and raise it to the power of the next entry.</td>
</tr>
<tr>
<td>$x!$</td>
<td>[ ! ]</td>
<td>Return the factorial of the current entry. Note that the factorial function is only valid for positive integers.</td>
</tr>
<tr>
<td>$\text{Cos}$</td>
<td>[ ^c ]</td>
<td>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).</td>
</tr>
<tr>
<td>$\text{Sin}$</td>
<td>[ ^s ]</td>
<td>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).</td>
</tr>
<tr>
<td>$\text{Tan}$</td>
<td>[ ^t ]</td>
<td>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).</td>
</tr>
<tr>
<td>$\ln$</td>
<td>[ N ]</td>
<td>Return the natural logarithm of the current entry.</td>
</tr>
<tr>
<td>$\log$</td>
<td>[ G ]</td>
<td>Return the base 10 logarithm of the current entry.</td>
</tr>
<tr>
<td>Rand</td>
<td>[ ? ]</td>
<td>Return a random number between 0.0 and 1.0.</td>
</tr>
</tbody>
</table>

**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 user | system | network -from_ascii | -to_ascii filename

[ -db_file db_filename ]

DESCRIPTION

ce_db_build builds a Classing Engine database from an ASCII description file or 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.

An example of the contents of an ASCII description file is:

```
{
  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>)
  )
}
```

modified 28 February 1992
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 "refto-Types" - an indication to the CE client that FNS_TYPE values are really TYPE_NAMEs that can be looked up in the 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, _, -. 

modified 28 February 1992
**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  user | system | network -from_ascii filename [-db_file db_filename]

DESCRIPTION  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.

The ASCII description for merge has the same syntax as that used for building a database from scratch (see ce_db_build(1)).

An example ASCII file is:

```{ 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>)
)
```
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 "refto-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 timezone ] [ −12 ] [ −24 ] [ −alarm setting ]
   [ −alarmtime hr:min ] [ −alarmcmd cmd ] [ −analog ] [ −digital ]
   [ +date ] [ −date ] [ −help ] [ −hourcmd cmd ] [ −name app-name ]
   [ −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 environ-
ments. 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 Inter-
face.

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 for-
mat. The clock window is scalable in either format.

OPTIONS  
−Wn Start clock with no title line.
+Wn Start clock with a title line present.
−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.
−TZ timezone
   Start the clock with an alternate time zone.
−12 If the digital clock is being displayed in the open window, then use a 12 hour
clock.
−24 If the digital clock is being displayed in the open window, then use a 24 hour
clock.
−alarm setting
   Set how often you want the alarm to go off. Choices are "none", "once" and
   "daily".
−alarmtime hr:min
   The time you want the alarm to go off.
−alarmcmd cmd
   The command you want to run, when the alarm does go off.
−analog
   Start up the open window display with the analog clock.
−digital
   Start up the open window display with the digital clock.
+date Show the current date in the open window.
−date Do not show the current date in the open window.
−help List 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 con-
   tain ‘.’ 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:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deskset.clock.faceAnalog</td>
<td>True, False (True)</td>
<td>Setting for the open clock face. Choices are analog and digital.</td>
</tr>
<tr>
<td>deskset.clock.iconAnalog</td>
<td>True, False (True)</td>
<td>Setting for the closed clock icon. Choices are analog and roman.</td>
</tr>
<tr>
<td>deskset.clock.iconHasWindowColor</td>
<td>True, False (False)</td>
<td>Determines whether the clock icon should use the window background color. By default, it uses the workspace color.</td>
</tr>
<tr>
<td>deskset.clock.digital12Hour</td>
<td>True, False (True)</td>
<td>Setting for digital displays. Choices are 12 hour and 24 hour display.</td>
</tr>
<tr>
<td>deskset.clock.showLocal</td>
<td>True, False (True)</td>
<td>Indicates whether the local or other timezone should be initially displayed.</td>
</tr>
<tr>
<td>deskset.clock.secondHand</td>
<td>True, False (False)</td>
<td>When True, a second hand (for analog) or a seconds value (for digital) will be displayed.</td>
</tr>
<tr>
<td>deskset.clock.date</td>
<td>True, False (False)</td>
<td>When True, the current date will be displayed.</td>
</tr>
<tr>
<td>deskset.clock.timeZone</td>
<td>Timezone (string)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>If present, the timezone to monitor.</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Resource:</strong></td>
<td>deskset.clock.alarmHrValue</td>
<td></td>
</tr>
<tr>
<td><strong>Values:</strong></td>
<td>Hour value (numeric)</td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
<td>The hour value for the alarm command.</td>
<td></td>
</tr>
</tbody>
</table>

| **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.

modified 24 March 1992
NAME  

cm – calendar manager, appointment and resource scheduling tool

SYNOPSIS  

```cm [generic-tool-arguments] [-c calendar] [-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  

cm is an appointment and resource scheduling tool that allows you to do the following:
- 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

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.


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:

```month/day holiday-name```

OPTIONS  

- `-c calendar` 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.

  generic-tool-arguments
  
  cm accepts the generic tool arguments described in xview(7).

- `-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.

USAGE  

cm operates via a set of pulldown menus from button stacks in a control panel. The menu commands are described below.

The Control Area

modified 9 Mar 1992
<table>
<thead>
<tr>
<th>View Menu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Day</strong></td>
<td>Displays the Day View. Also displays the Appointments and Todo items in a window for the selected day.</td>
</tr>
<tr>
<td><strong>Week</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Month</strong></td>
<td>Displays the Month View. Also displays the Appointments and Todo items in a window for the month.</td>
</tr>
<tr>
<td><strong>Year</strong></td>
<td>Displays a year calendar and the appointments and Todo items for a year.</td>
</tr>
<tr>
<td><strong>Time Zone</strong></td>
<td>Sets time zones. The current times are changed to reflect the new time zone.</td>
</tr>
<tr>
<td><strong>Find...</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Go To...</strong></td>
<td>Displays the calendar at the date requested.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Edit Menu</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Appointment</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Date:</strong></td>
<td>The date is automatically set by the tool when a day is selected. You may change the date format through the property settings.</td>
</tr>
<tr>
<td><strong>Start:</strong> and <strong>Stop:</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>What:</strong></td>
<td>Filled in with information about the appointment. This information will be displayed on the various views.</td>
</tr>
<tr>
<td><strong>Alarm:</strong></td>
<td>Determines what action will be taken in advance of the appointment.</td>
</tr>
<tr>
<td><strong>Beep</strong></td>
<td>Produces an audible sound.</td>
</tr>
<tr>
<td><strong>Flash</strong></td>
<td>Inverts the frame of the tool or the icon if the tool is closed.</td>
</tr>
<tr>
<td><strong>Popup</strong></td>
<td>Pops up a reminder window with the text of the appointment showing.</td>
</tr>
</tbody>
</table>
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

cmdtool – run a shell (or other program) in an OpenWindows enhanced terminal window

SYNOPSIS

```
  cmdtool [ −C ] [ −M bytes ] [ −P count ] [ −B boldstyle ] [ −I command ]
  [ generic-tool-arguments ] [ program [ program-arguments ] ]
```

DESCRIPTION

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).

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.

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:

```
  cmdtool.maxLogFileSize: max_size
```

OPTIONS

−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.

−M bytes  Set the log file /tmp/tty.txt.pid to wrap-around after the indicated number of bytes 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 bytes.

−P count  Checkpoint the log after every set of count editing operations.

−B boldstyle  Set the style for displaying bold text to boldstyle. boldstyle 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.

−I command  Pass command to the shell. SPACE characters within the command must be escaped.

generic-tool-arguments

Cmdtool accepts the generic tool arguments listed in xview(7).

modified 30 November 1993
program [ program-arguments ]
If a program argument is present, cmdtool runs it and passes any
remaining arguments to that program. If no program is given, cmdtool
 runs the program indicated by the SHELL environment variable, or
/bin/sh by default.

**USAGE**

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 keyboard. See xview(7) for more detailed information.

**cmdtool Windows**

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.

**cmdtool Windows**

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.

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 ttty window (as with shelltool(1)). When that program terminates or sleeps, the full command window functionality is restored.

**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.**

**The cmdtool Menu**

The cmdtool window menu is called the Term Pane menu and contains the following options and their submenus:

**History**

Creates a list of commands used during the cmdtool session.

**Mode**

**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.

**Edit**

Provides a set of editing functions for this window.

**Again**

Repeats the last action.

**Undo Last Edit**

Undoes the last edit made in cmdtool.

**Undo All Edits**
Undo all edits during this session of **cmdtool**.

**Copy**
Makes a copy of the selected text and places it on the clipboard.

**Paste**
Fastes 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

modified 30 November 1993
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.

**Accelerators**

Text facility accelerators that are especially useful in command windows are described here. See `textedit(1)` for more information.

- **CTRL-RETURN** Position the caret at the bottom, and scroll it into view as determined by `Text.LowerContext`.
- **CAPS-lock** Toggle between all-upper-case keyboard input, and mixed-case.

**FILES**

```
/tmp/tty.txt.pid   log file
~/.textswrc
~/.ttyswrc
/usr/openwin/lib/locale/C/xview/.text_extras_menu
$HOME/.TextExtraMenu
/etc/terminfo
/usr/bin/sh
```

**SEE ALSO**

`rlogin(1)`, `script(1)`, `shelltool(1)`, `textedit(1)`, `vi(1)`, `xview(7)`, `curses(3X)`

*Solaris User's Guide*

**BUGS**

Typing ahead while `cmdtool` changes between its scrollable and cursor motion modes will sometimes freeze `cmdtool`.

Full terminal emulation is not complete. Some manifestations of this deficiency are:

- 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 calendar ] [ -d date ] [ -v view ]

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 date, view 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

- 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.

- d date 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.

- 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".

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_delete has no arguments:

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

modified 9 March 1992
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)?

text% cm_delete

To delete at a specific date:

text% 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)?

text% cm_delete

To delete from a specific target calendar:

text% 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)?

eexample%

To delete multiple appointments:

eexample% 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)?

eexample%
cm_delete(1) User Commands OpenWindows Desktop 3.5

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

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

modified 9 March 1992
NAME

cm_insert – insert appointments into Calendar Manager database

SYNOPSIS

cm_insert [ -c calendar ] [ -d date ] [ -s start ] [ -e end ] [ -v view ] [ -w what ]

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

- 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.

- d date 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.

- s start 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.

- e end 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.

- 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".

- w what What the appointment is about. The user may specify a string of text describ-
ing 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".

modified 9 March 1992
**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:

```bash
eexample% 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

eexample%

Lastly, for multiple line appointments:

```bash
eexample% 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

eexample%

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)

modified 9 March 1992
NAME  

cm_lookup – look up appointments from Calendar Manager database

SYNOPSIS  

cm_lookup [ -c calendar ] [ -d date ] [ -v view ]

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  

-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.

-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.

-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".

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_lookup has no arguments:

test% 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

test%
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
```

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
```

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  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  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.

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.

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)

Solaris User’s Guide
**NAME**
convert_to_Xdefaults – convert SunView1 defaults into equivalent Xdefaults

**SYNOPSIS**
convert_to_Xdefaults [ filename ]

**AVAILABILITY**
Available with the OpenWindows Application Environment. 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 SunView1.x defaults to X Window (Xdefaults) defaults. convert_to_Xdefaults reads filename, converting SunView1 defaults into their equivalent Xdefaults for XView. Defaults that are no longer supported or are not recognized as standard SunView1 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 $HOME directory and should be named .Xdefaults.

**EXAMPLES**
For an example of how this command works, run the script on your Sunview1 defaults file:

convert_to_Xdefaults $HOME/.defaults > $HOME/.Xdefaults

When the script is complete, edit the resulting file ( $HOME/.Xdefaults ) and remove any unconverted entries and/or make modifications suggested in the file by the script. Remove the comments and the ! sign.

**SAMPLES**
SunView1 defaults entries ... 

```
/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"
```

are converted to the Xdefault...  

```
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 milliseconds) 100
```
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 ] filename...

AVAILABILITY
This command is available with the XView software distribution.

DESCRIPTION
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
filename and creates a new file with the XView API in the current directory called
filename.converted_to_xview. The default conversion that is done is called FULL conver-
sion. FULL conversion of SunView source converts everything to XView naming conven-
tions regardless of API compatibility support (e.g., changes WIN_FONT to XV_FONT
even though WIN_FONT would still work).

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:

    #ifdef XVIEW_COMMENT
    XView CONVERSION - Make sure to use xv_init to process the attrs
    first. Sect 3.2
    #endif

The original SunView file is not modified. After the file is converted, you should then
search for

    XView CONVERSION

in the new converted program (filename.converted_to_xview). Use the conversion docu-
mentation, XView 3.1: Converting SunView Applications, to determine the proper conver-
sion 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.

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
Convert foo.c from SunView1 to XView:

    % convert_to_xview foo.c
    ----Converting File: foo.c
    --Done
    %

modified 30 November 1993
Now go in and edit (with your favorite text editor such as \texttt{vi}, \texttt{textedit}, etc.) the result of the conversion (\texttt{my\_program.c\_converted\_to\_xview}) and see if there is anything that didn’t get converted:

\texttt{% textedit foo.c\_converted}

Do only minimal conversion of \texttt{my\_program.c} & \texttt{your\_program.c} to XView:

\texttt{% convert\_to\_xview -m foo.c blah.c}
\texttt{----Converting File: foo.c}
\texttt{----Converting File: blah.c}
\texttt{--Done}
\texttt{%}

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

\textbf{FILES}

\texttt{$OPENWINHOME/bin/convert\_to\_xview$

Where \texttt{$OPENWINHOME$} is the installation/mount point for XView.

\textbf{SEE ALSO} \texttt{sed(1), textedit(1), vi(1), sh(1)}

\par\vfill\n
\small{modified 30 November 1993}
<table>
<thead>
<tr>
<th>NAME</th>
<th>docviewer – OpenWindows viewer for AnswerBook on-line documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td><code>docviewer -d document-name -p tooltalk-procid [ -c card-catalog-file ]</code></td>
</tr>
<tr>
<td>AVAILABILITY</td>
<td>SUNWowrqd</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td><code>docviewer</code> and <code>navigator</code>(1) constitute an OpenWindows application for viewing and navigating AnswerBook on-line document collections. <code>docviewer</code> displays PostScript files found using <code>navigator</code>. <code>docviewer</code> lets you page through a document, print pages, magnify or reduce pages, and follow hypertext links and cross-references to other documents. You should not start <code>docviewer</code> directly. It is started automatically by <code>navigator</code> as needed. If you are running several navigators simultaneously, they will each start their own docviewers.</td>
</tr>
</tbody>
</table>
| OPTIONS     | `docviewer` accepts most of the generic tool arguments described in `xview(7)`, as well as the following options:  
  - `-d document-name`  
    Specify the name of the document to be viewed.  
  - `-c card-catalog`  
    Specify the name of the card catalog file used to locate AnswerBooks. See `ab_cardcatalog(4)` for more information.  
  - `-p tooltalk-procid`  
    Specify the tooltalk process id of the navigator invoking this docviewer. |
| USAGE       | Once a document is displayed in the `docviewer` window, you can view it as follows:  
  **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.  
  **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.  
  **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. |
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      dsdm – Drop Site Database Manager

SYNOPSIS   dsdm [ −x ]

DESCRIPTION  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.

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.

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.

OPTIONS    
−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 managing the drop site database, it will stop maintaining the database.

DIAGNOSTICS  another DSDM is already running
Another client has ownership of the _SUN_DRAGDROP_DSDM selection.

SEE ALSO   olwm(1)
<table>
<thead>
<tr>
<th>NAME</th>
<th>ds_server_init – store a property in the X11 server to reduce colormap flashing within the DeskSet applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>ds_server_init [ -f filename ] [ -a ]</td>
</tr>
<tr>
<td>AVAILABILITY</td>
<td>Available with the OpenWindows environment.</td>
</tr>
</tbody>
</table>
| DESCRIPTION | 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.

The property is stored as a character string under the name _SUN_DESKSET_COLORS 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 $OPENWINHOME/share/xnews/client/ds_server_init/ds_colors.txt. 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.

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 $HOME/.Xdefaults file and are 1) Window.ForegroundColor, 2) Window.BackgroundColor, 3) OpenWindows.WorkspaceColor, and 4) OpenWindows.WindowColor.

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. |
| INSTALLATION | ds_server_init is invoked when the X11 server is started. |
| OPTIONS | 
| -f filename | Indicates the filename containing the colors. $OPENWINHOME/share/xnews/client/ds_server_init/ds_colors.txt is the default file. |
| -a | 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. |
| FILES | 
| $OPENWINHOME/share/xnews/client/ds_server_init/ds_colors.txt | This file contains the set of DeskSet colors in ASCII where the ds_server_init utility reads the colors from. |
| $HOME/.Xdefaults | modified 17 March 1992 |
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 $OPENWINHOME/share/src/tooltalk/demo/edit_demo. Your OPENWINHOME environment variable must be set to where OpenWindows is installed (typically /usr/openwin). 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.

SEE ALSO 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 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**

filemgr – a GUI-based file management application

**SYNOPSIS**


**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**

In addition to the generic tool arguments described in xview(7), filemgr can accept the following options:

- `-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 directory`
  This will start filemgr in the given directory.

- `-i secs` This will set the timer to check on folder and/or file modification times every secs seconds.

- `-M` Just display using the foreground and background colors only, even on a color screen.

- `-name app-name`
  This option instructs filemgr to use resources associated with app-name in your $HOME/.desksetfdefaults file instead of using resources for filemgr. This lets you run multiple filemgrs at the same time with different attributes. app-name 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 expresion.

**RESOURCES**

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

modified 2 December 1993
<table>
<thead>
<tr>
<th>Resource:</th>
<th>applicationColor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>Color name string or hexadecimal color specification string</td>
</tr>
<tr>
<td>Description:</td>
<td>The color of the default application icon.</td>
</tr>
<tr>
<td>Resource:</td>
<td>autoShowCD</td>
</tr>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description:</td>
<td>If set to true, when a CD is inserted, then filemgr will display it’s contents in a new window.</td>
</tr>
<tr>
<td>Resource:</td>
<td>autoSortOnUpdate</td>
</tr>
<tr>
<td>Values:</td>
<td>True, False (False)</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
<tr>
<td>Resource:</td>
<td>cacheSize</td>
</tr>
<tr>
<td>Values:</td>
<td>0-100000000 (15000000)</td>
</tr>
<tr>
<td>Description:</td>
<td>Determines the limit (in bytes) for the garbage collection run by the fmgc(1) program (started by filemgr). fmgc will continue to prune the &quot;/.fm hierarchy until this limit is reached.</td>
</tr>
<tr>
<td>Resource:</td>
<td>cdromContentMatch</td>
</tr>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
<tr>
<td>Resource:</td>
<td>classingEngineBufferSize</td>
</tr>
<tr>
<td>Values:</td>
<td>0-32768 (512)</td>
</tr>
<tr>
<td>Description:</td>
<td>Denotes how many bytes are read from each file, and passed onto the Classing Engine to try to determine it’s file type.</td>
</tr>
<tr>
<td>Resource:</td>
<td>confirmDelete</td>
</tr>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description:</td>
<td>When true, sends deleted files to the wastebasket. Otherwise, files are deleted normally as if you had run a rm(1) command.</td>
</tr>
<tr>
<td>Resource:</td>
<td>confirmDeleteFolder</td>
</tr>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
<tr>
<td>Resource</td>
<td>confirmDestroyFolder</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>confirmQuit</th>
<th>Values: True, False (True)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>When true, the user will be prompted when they select the &quot;Quit File Manager&quot; option from the File button menu, to ask if they really want to do this. Setting this resource to false disables this prompt.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>documentColor</th>
<th>Values: Color name string or hexadecimal color specification string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The color of the default application icon.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>fileCheckInterval</th>
<th>Values: 0-9999 (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The amount of time (in seconds) between checks on the various monitored file systems to see if changes have occurred.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>filenameNoChars</th>
<th>Values: 0-255 (255)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>filterScript</th>
<th>Values: regular expression (NULL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A regular expression describing which files you wish to view. The default is empty, which indicates all files should be displayed.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>floppyContentMatch</th>
<th>Values: True, False (True)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>folderColor</th>
<th>Values: Color name string or hexadecimal color specification string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The color of the default folder icon.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>followSymbolicLinks</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource:</td>
<td>Values:</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>iconDirectionVertical</td>
<td>True, False (False)</td>
<td>When false, displays files row-by-column. Otherwise, displays files column-by-row.</td>
</tr>
<tr>
<td>iconListStyle</td>
<td>int (0)</td>
<td>Saves the state of the &quot;List Options&quot; toggles on the Customize View popup. This is not very human readable and should be changed in the future.</td>
</tr>
<tr>
<td>loadDirectoryState</td>
<td>True, False (True)</td>
<td>When false, directory state information for each directory will not be read from or written to the users’ ~/.fm directory.</td>
</tr>
<tr>
<td>loadIconPositions</td>
<td>True, False (True)</td>
<td>When false, icon positional information for each directory will not be read from or written to the users’ ~/.fm directory. Setting this resource to true, overrides the loadDirectoryState resource.</td>
</tr>
<tr>
<td>maxGotoMenuEntries</td>
<td>1-500 (10)</td>
<td>The maximum number of entries that will be displayed in the Goto button menu.</td>
</tr>
<tr>
<td>newFolderName</td>
<td>String (NewFolder)</td>
<td>The initial name given to newly created folders.</td>
</tr>
<tr>
<td>newWindowOnDirectoryOpen</td>
<td>True, False (False)</td>
<td>Whether a new popup sub-folder pane is created when a folder is opened.</td>
</tr>
<tr>
<td>otherEditor</td>
<td>UNIX command (shelltool sh -c &quot;sleep 3; vi $FILE&quot;)</td>
<td>The open method the user wishes to use for opening files. Note this is modified 2 December 1993</td>
</tr>
</tbody>
</table>
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 (cat $FILE | mp -lo | lp) |
| Description | The print method the user wishes to use for printing files. |

| Resource: | shellToolName |
| Values: | UNIX command (cmdtool) |
| 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). |

<p>| Resource: | treeIconYPosition |
| Values: | 0-&quot;Height of Screen&quot; (0) |
| Description | Saves the Y location of the folder view window icon (in pixels). |</p>
<table>
<thead>
<tr>
<th>Resource:</th>
<th>treePaneGap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>48-320 (96)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>treeView</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (False)</td>
</tr>
<tr>
<td>Description</td>
<td>When True, display the folder view window on startup.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>treeWidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>0-“Width of Screen” (40 characters)</td>
</tr>
<tr>
<td>Description</td>
<td>Saves the width of the open folder view window (in pixels).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>treeWindowXPosition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>0-“Width of Screen” (0)</td>
</tr>
<tr>
<td>Description</td>
<td>Saves the X location of the open folder view window (in pixels).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>treeWindowYPosition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>0-“Height of Screen” (0)</td>
</tr>
<tr>
<td>Description</td>
<td>Saves the Y location of the open folder view window (in pixels).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>useCache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description</td>
<td>When False, filemgr will not read or write directory cache information into the users “/.fm” hierarchy.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>useClassingEngine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>useTextedit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description</td>
<td>When True, use the texteditor to open files. Otherwise, use the user defined open method described above under “otherEditor”.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>viewType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>Icon, List, Content (Icon)</td>
</tr>
<tr>
<td>Description</td>
<td>Saves the state of the &quot;Display Mode&quot; toggles on the Customize View popup.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>wastebasketClosed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
</tbody>
</table>

modified 2 December 1993
**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!

modified 2 December 1993
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 < frame_file > output_file

**DESCRIPTION**
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. fixframe is automatically run by pageview when a user attempts to view a FrameMaker 2.0 POSTSCRIPT file using pageview.

Specifically, fixframe saves font definitions that occur between pages and inserts them after the %%Page: comment for each page.

**SEE ALSO**
pageview(1)


**TRADEMARK**
POSTSCRIPT is a registered trademark of Adobe Systems Inc.
<table>
<thead>
<tr>
<th><strong>NAME</strong></th>
<th>fixinterleaf – convert Interleaf PostScript files to conform to the PostScript structuring conventions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SYNOPSIS</strong></td>
<td>fixinterleaf interleaf_file output_file</td>
</tr>
<tr>
<td><strong>DESCRIPTION</strong></td>
<td>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. fixinterleaf is automatically run by pageview when a user attempts to view an Interleaf POSTSCRIPT file using pageview. 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.</td>
</tr>
<tr>
<td><strong>SEE ALSO</strong></td>
<td>pageview(1)</td>
</tr>
</tbody>
</table>

POSTSCRIPT is a registered trademark of Adobe Systems Inc
NAME  

fmgc – a program to garbage collect a users’ .fm directory hierarchy

SYNOPSIS  


DESCRIPTION  

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.

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.

OPTIONS  

- l limit  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.

- u username  
  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.

- v  This will display the current version number of fmgc.

- V  Be verbose. Output messages stating what’s happening.

- ?  This will display a list of command line options which can be passed to fmgc.

SEE ALSO  

filemgr(1),

FILES  

$HOME/.fm  
This directory contains directory state, icon positional and file status information saved by the filemgr program.

$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.

modified 5 January 1993
NAME

format_floppy – diskette formatting program used by FileManager

SYNOPSIS

format_floppy -d device -m mnt_point [ -h ] [ -n name ] [ -p popup_type ] [ -r ] [ -x x_pos ] [ -y y_pos ]

DESCRIPTION

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 cannot 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.

Note that DOS-formatted disks are automatically recognized by vold and filemgr will not try to reformat them.

OPTIONS

- d device 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 N is the number of hte floppy disk (usually 0).

- h Causes format_floppy to display its "help" or usage message.

- m mnt_point The mount-point of the floppy disk must be specified. This information is in the first field of the the file /tmp/.removable/floppyN, where N is the number of the floppy disk (usually 0).

- n name Allows a name to be assigned to the newly-formatted floppy disk.

- p popup_type Allows the type of popup menu to be specified where popup_type can be format, unformatted, or unlabeled. The default is format. Note that the wording in these popup menus is different.

- r Allows the floppy disk to be renamed. If -r is specified, -n must be also.

- x x_pos Used with the -p option to specify the x-coordinate position for the popup menu.

- y y_pos Used with the -p option to specify the y-coordinate position for the popup menu.

FILES

/tmp/.removable/*

SEE ALSO

filemgr(1), vold(1)

modified 30 November 1993
NAME
helopen – utility for programmatically controlling helpviewer

SYNOPSIS
helopen help-handbook-file

DESCRIPTION
helopen is a utility for programmatically controlling helpviewer(1).

helopen sends a ToolTalk request to a running helpviewer to display the specified help
handbook file. If no helpviewer is running, helopen starts one.

OpenWindows users do not typically need to use helopen directly. It is used to imple-
ment the "more help" feature in OpenWindows spot help.

OPTIONS
helopen accepts the following command line options:

- \( -f \) help-handbook-file
  Specify the name of the help handbook to be viewed. The file name must be rela-
tive to a directory in $HELPPATH. See helpviewer(1) for more information.

DIAGNOSTICS
helopen prints an error message on failure and returns non-zero exit status.

BUGS
helpviewer and helopen 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
           −f help-handbook-file

DESCRIPTION  helpviewer is an OpenWindows application for viewing and navigating on-line help handbooks.

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.

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.

OPTIONS  helpviewer accepts most of the generic tool arguments described in xview(7), as well as the following options:

−f help-handbook-file
   Specify the name of the help handbook to be viewed. This file name should be specified relative to a directory in $HELPPATH (see ENVIRONMENT below).

USAGE  Once a help handbook is displayed in the helpviewer window, you can view it as follows:

Page Turning
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.

Page History
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.

Hypertext Links
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.

Page Magnification
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"
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 `$HELPPATH` variable to locate handbook files. `$HELP_PATH` 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**

iconedit – create and edit images for OpenWindows icons, cursors, and panel items

**SYNOPSIS**

```
iconedit [ filename ] [ XView command-line arguments ]
```

**AVAILABILITY**

This command is available with the OpenWindows user environment. For information about XView command-line arguments see the XView documentation.

**OPTIONS**

- **filename** Contains the image.
- **XView command-line arguments**
  - `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.

modified 20 Feb 1992
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 ] [ imagefile ]
```

**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, jif (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:

<table>
<thead>
<tr>
<th>File</th>
<th>The File menu contains the following items.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open...</td>
<td>Brings up a dialog which allows the user to select a file that is to be opened for viewing.</td>
</tr>
<tr>
<td>Open As...</td>
<td>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.</td>
</tr>
<tr>
<td>Save...</td>
<td>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.</td>
</tr>
<tr>
<td>Save As...</td>
<td>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.</td>
</tr>
<tr>
<td>Save Selection As...</td>
<td>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.</td>
</tr>
<tr>
<td>Save Page As Image...</td>
<td>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 if viewing a page of a multipage file.</td>
</tr>
<tr>
<td>Print One</td>
<td>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.</td>
</tr>
<tr>
<td>Print Preview...</td>
<td>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.</td>
</tr>
<tr>
<td>Print...</td>
<td>Brings up a dialog which allows user to set various parameters</td>
</tr>
</tbody>
</table>

modified 25 March 1994
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**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>−v</td>
<td>Prints out the current version of <code>imagetool</code>.</td>
</tr>
<tr>
<td>−verbose</td>
<td>Prints lots of debugging information (not useful to the user)</td>
</tr>
<tr>
<td>−timeout seconds</td>
<td>Set the timeout value for the Display POSTSCRIPT server. The default value is 60 seconds.</td>
</tr>
<tr>
<td>−usage</td>
<td>Prints out valid command line options.</td>
</tr>
</tbody>
</table>

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`.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViewImageIn</td>
<td>Color, GrayScale (Color)</td>
<td>Determines if images will be displayed in color or grayscale. If user is using a monochrome monitor, this setting has no affect.</td>
</tr>
<tr>
<td>Colors</td>
<td>BW, 16, 256, Millions (256)</td>
<td>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.</td>
</tr>
<tr>
<td>DisplayPalette</td>
<td>True, False (True)</td>
<td>Determines if palette is automatically displayed when first image is opened by <code>imagetool</code>.</td>
</tr>
</tbody>
</table>
| UseDSC | True, False (False) | 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

modified 25 March 1994
<table>
<thead>
<tr>
<th>NAME</th>
<th>locale_env - program for openwin-sys</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>locale_env</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>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.</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>openwin(1)</td>
</tr>
</tbody>
</table>

modified 12 Nov 1992
NAME
mailp, digestp, filep, newsp, filofaxp, franklinp, timemanp, timesysp – frontends to the
mp PostScript pretty printer

SYNOPSIS
mailp [options] filename ...
newsp [options] filename ...
digestp [options] filename ...
filep [options] filename ...
filofaxp [options] filename ...
franklinp [options] filename ...
timemanp [options] filename ...
timesysp [options] filename ...

DESCRIPTION
mailp is a frontend to the mp(1) Postscript pretty printer program. It uses different
names to provide various mp options:
mailp will print out mail messages.
newsp will print out USENET news articles.
digestp will print out USENET digest files.
filep will print out ordinary ASCII files.
filofaxp will print out in Filofax personal organiser format.
franklinp will print out in Franklin Planner personal organiser format.
timemanp will print out in Time Manager personal organiser format.
timesysp will print out in Time/System International personal organiser format.
mailp (and the associated programs) read each filename in sequence and generate a
prettified version of the contents in PostScript format. If no filename arguments are pro-
vided, mailp reads the standard input.
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 ps printer.

OPTIONS
-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 printer Send output to the named printer. Otherwise send output to the printer named in
the PRINTER environment variable,
-h Banner printing is disabled. Most of the information that typically appears on the
banner sheet is output in the mp banners.
-d printer Send output to the named printer. Otherwise send output to the printer named in
the PRINTER environment variable,
−I 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 subject will default to the name of each of these files.

SEE ALSO  mp(1)

Revisited by Larry W. Virden - August, September 1990.
Revisited by Bruno Pillard - September 1990.
NAME
mailprint – filter to strip out mail message attachments

SYNOPSIS
mailprint [-b] [filename...]

DESCRIPTION
The mailprint program reads a filename (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 filename argument is provided, mailprint reads from the standard input.

mailprint will add the following line to the end of each mail message, if any attachments were stripped out:

This message contains n attachments (not printed)
where n is the number of attachments stripped.

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).

mailprint can be used together with other programs to print out a message. For example:

example% cat filename | mailprint | mp | lp

OPTIONS
- 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 ] [ generic-tool-arguments ] [ -v ]

AVAILABILITY

This command is available with the OpenWindows environment. mailtool uses the OPEN LOOK Graphical User Interface.

DESCRIPTION

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. The full editing capabilities of textedit(1) are available for modifying and composing mail messages as well as text fields within command panels.

OPTIONS

-Mx Expert mode. Do not ask for confirmation after potentially damaging mail commands. This has the same effect as setting the expert variable.

-Mi interval

Check for new mail every interval seconds. This has the same effect as specifying a value for the interval variable.

-Mf mailfile

Start up mailtool with mailfile loaded instead of the In-Box.

-v

Print out the version number of mailtool and exit.

generic-tool-arguments

Mailtool accepts the generic tool arguments described in xview(7).

USAGE

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.

Command Panel Buttons

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.

File >

Load In-Box [Meta o]

Used to read mail from the user’s system mail file into mailtool.

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
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.

modified 7 January 1992
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: ".

*modified 7 January 1992*
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.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expert</td>
<td>Set expert mode in which minimal confirmations are requested. This variable can be accessed through</td>
</tr>
<tr>
<td></td>
<td>the Request confirmations check box in the Compose Window Category of the Mail Tool Property Sheet.</td>
</tr>
<tr>
<td>filemenu</td>
<td>A list of files from which to initialize the Move, Copy, and Load menus. These can be absolute</td>
</tr>
<tr>
<td></td>
<td>pathnames or pathnames relative to the directory specified in the folder variable. This variable is</td>
</tr>
<tr>
<td></td>
<td>superceded by the filemenu2 variable.</td>
</tr>
<tr>
<td>filemenu2</td>
<td>Same as filemenu, but if both exist, filemenu2 takes precedence. This variable can be accessed</td>
</tr>
<tr>
<td></td>
<td>through the Move, Copy, Load Menus: scrolling list and the Permanent File: text field in the Mail</td>
</tr>
<tr>
<td></td>
<td>Filing category of the Mail Tool Property Sheet.</td>
</tr>
<tr>
<td>filemenusize</td>
<td>Specifies the maximum number of entries in the Move, Copy, and Load menus. This variable can be</td>
</tr>
<tr>
<td></td>
<td>accessed through the Display Up To: __ Files in Menus entry in the Mail Filing Category of the Mail</td>
</tr>
<tr>
<td></td>
<td>Tool Property Sheet. The default is 10.</td>
</tr>
<tr>
<td>flash</td>
<td>The number of times to flash the window or icon when new mail arrives. This variable can be</td>
</tr>
<tr>
<td></td>
<td>accessed through the Mail Tool Property Sheet in the Header Window Category as Signal With: __</td>
</tr>
<tr>
<td></td>
<td>Flash(es). The default is 0.</td>
</tr>
<tr>
<td>folder</td>
<td>The directory for saving mail files. This variable can be accessed as Mail File Directory in the Mail</td>
</tr>
<tr>
<td></td>
<td>Filing category of the Mail Tool Property Sheet.</td>
</tr>
<tr>
<td>headerlines</td>
<td>The number of lines to display at a time in the header window. This variable can be accessed through</td>
</tr>
<tr>
<td></td>
<td>the Mail Tool Property Sheet in the Header Window category as Display __ Headers. The default is 15.</td>
</tr>
<tr>
<td>hideattachments</td>
<td>Hide the attachments pane in the Compose Message window. This variable can be accessed through the</td>
</tr>
<tr>
<td></td>
<td>Defaults: Show attachment list check box in the Compose Window category of the Mail Tool Property Sheet.</td>
</tr>
<tr>
<td></td>
<td>The default is to show the attachment pane.</td>
</tr>
<tr>
<td>indentprefix</td>
<td>When indentprefix is set, the string that it is set to is used to mark indented lines from included</td>
</tr>
<tr>
<td></td>
<td>messages. The default indentprefix is “&gt; “.</td>
</tr>
<tr>
<td>popuplines</td>
<td>The number of lines in the View Message and Compose Message Windows. This variable can be accessed</td>
</tr>
<tr>
<td></td>
<td>through the Mail Tool Property Sheet in the Message Window category as Display __ Lines of Text. The</td>
</tr>
<tr>
<td></td>
<td>default is 30.</td>
</tr>
<tr>
<td>printmail</td>
<td>The command to use to print a message. This variable can be accessed through the Mail Tool Property</td>
</tr>
<tr>
<td></td>
<td>Sheet in the Message Window category as Print Script. The default is lp -s.</td>
</tr>
<tr>
<td>record</td>
<td>The mail file in which to record outgoing messages. If record is set, a Log check box will appear on</td>
</tr>
<tr>
<td></td>
<td>the Compose Message window. If the check box is</td>
</tr>
</tbody>
</table>
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 Win-
dow 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 (e.g., 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 emp-
tied 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**

modified 7 January 1992
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:

```plaintext
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.
- `~/mtdelete.log` 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

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] [−?] [filename...]

DESCRIPTION

The mp program reads each filename in sequence and generates a prettyfied 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.

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.

The program is conveniently used in conjunction with the print button of the mailtool(1) program, or the pipe command provided by mail(1). Add the following two lines to your .mailrc file:

```
set printmail='mp | lp'
set cmd="mp | lp &"
```

Source the .mailrc file, and you are ready to use mp. For printing ordinary ASCII files, the following alias (to be placed in your

```
alias print 'mp -o -s "\!*" <\!* | lp'
```

OPTIONS

−A4 Use A4 paper size (8.5 x 11.4 inches).

−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.

−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.

−L localename

Provide the locale of the file to be printed. If this command line option is not present, then mp looks for the MP_LANG environment variable. If that is not present, then the LANG 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.

−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.

−US Use US paper size (8.5 x 11 inches). This is the default paper size.

−a Format the file as a news article. The top banner contains the text: "Article from
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

modified 14 June 1993
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 csh(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), mailtool(1),

**AUTHORS**

Original version by Steve Holden.
Converted to C, modified and maintained by Rich Burridge, SunSoft Inc, Mountain View.
Original modified to handle net news articles and MH mail by Bruno Pillard, Chorus Systems, France.
Handling of mail digests added by Dave Glowacki of Public Works Computer Services, St Paul, MN.

modified 14 June 1993
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 Burridge. MAIL: richb@Eng.Sun.COM
NAME  
navigator – browse and search AnswerBook on-line documentation

SYNOPSIS  
navigator [ −b library-file ] [ −c card-catalog ]

DESCRIPTION  
navigator and docviewer(1) together comprise an OpenWindows application for viewing
and navigating AnswerBook on-line document collections.
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).

Documents found using navigator may then be viewed with docviewer(1).

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.

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.

OPTIONS  

−b library-file
Specify the AnswerBook library to load. The default library is
$HOME/.ab_library. See ab_library(4) for more information.

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

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.

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.

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.

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.

Search Mode
Click SELECT on the "Search" button near the top of the Navigator window to

modified 18 November 1993
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*`

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  olbiff – OLIT mailbox monitor


DESCRIPTION  Olbiff monitors the user’s mail spool file at a specified interval, and displays information about incoming messages.

OPTIONS  
- show             display n lines in ListBox
- interval         check mail every n seconds
- fl n              display no more than n characters for name
- sl n              display no more than n 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  xbbiff(1)
<table>
<thead>
<tr>
<th>NAME</th>
<th>olitsampler – OLIT widget demo program</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>olitsampler</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Olitsampler is a simple application program that demonstrates OLIT widgets.</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>OLIT Reference Manual</td>
</tr>
<tr>
<td><strong>NAME</strong></td>
<td>olittable – OLIT widget demo program</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td><strong>SYNOPSIS</strong></td>
<td>olittable</td>
</tr>
<tr>
<td><strong>DESCRIPTION</strong></td>
<td>Olittable is a simple application program that demonstrates OLIT widgets laid out in an imitation periodic table. 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. By default, olittable looks in the directory $OPENWINHOME/lib/help/olittable, then in the user’s current working directory.</td>
</tr>
<tr>
<td><strong>ENVIRONMENT</strong></td>
<td>TABLEINFO – overrides the location of the *.txt files containing the auxiliary widget information.</td>
</tr>
<tr>
<td><strong>FILES</strong></td>
<td>*.txt each widget has a file whose base filename is the abbreviated symbol used in the periodic table (&quot;Rb&quot; for RectButton, or &quot;Cp&quot; for Caption.)</td>
</tr>
<tr>
<td><strong>SEE ALSO</strong></td>
<td>OLIT Reference Manual</td>
</tr>
</tbody>
</table>

modified 19 July 91
<table>
<thead>
<tr>
<th>NAME</th>
<th>olmh – OpenLook interface to the MH message handling system</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td><code>olmh [ -path mailpath ] [ -initial foldername ] [ -flag ] [ -toolkitoption... ]</code></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>The <code>olmh</code> program provides a window-oriented user interface to the Rand MH Message Handling System. To actually do things with your mail, it makes calls to the MH package. Electronic mail messages may be composed, sent, received, replied to, forwarded, sorted, and stored in folders. To specify an alternate collection of mail folders in which to process mail, use <code>-path</code> 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 MH Path is not given. To specify an alternate folder which may receive new mail and is initially opened by <code>olmh</code>, use the <code>-initial</code> flag. The default initial folder is ‘inbox’. The option <code>-flag</code> will cause <code>olmh</code> 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). 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. <code>olmh</code> really is easy to use.</td>
</tr>
<tr>
<td>INSTALLATION</td>
<td>The current version of <code>olmh</code> requires that the user is already set up to use MH, 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 MH; to convert to version 6, you must remove that line. (Failure to do so causes spurious output to stderr, which can hang <code>olmh</code> depending on your setup.) 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 <code>olmh</code> to incorporate new mail. For more information, refer to the mh(1) documentation.</td>
</tr>
<tr>
<td>BASIC SCREEN LAYOUT</td>
<td><code>olmh</code> starts out with a single window, divided into four main areas:</td>
</tr>
<tr>
<td></td>
<td>- Five buttons with pull-down command menus.</td>
</tr>
<tr>
<td></td>
<td>- A collection of buttons, one for each top level folder. New users of mh will have two folders, “drafts” and “inbox”.</td>
</tr>
<tr>
<td></td>
<td>- A listing, or Table of Contents, of the messages in the open folder. Initially, this will show the messages in “inbox”.</td>
</tr>
</tbody>
</table>
olmh uses the X Toolkit Intrinsics 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.

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.

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

modified 24 March 1994
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.
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

6 modified 24 March 1994
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
   RubberTile
   Control
     menuBox
       MenuButton folderButton
     MenuShell
       Form
         menu_form
           Control pane
             OblongButton open
             OblongButton openInNew
             OblongButton create
             OblongButton delete
             OblongButton close

   MenuButton tocButton
```

modified 24 March 1994
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

modified 24 March 1994
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

modified 24 March 1994
The hierarchy of the pick window:

Not implemented in this demo version of Olmh

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 Intrinsics 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...
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 mes-
sages 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 wait-
ing 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 MailWaiting-
Flag on.

MakeCheckpoints
If true, olmh will attempt to save checkpoints of volatile information. The fre-
quency 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".
<table>
<thead>
<tr>
<th><strong>PickGeometry</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial geometry for pick windows.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PointerColor</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The foreground color of the pointer. Default is XtDefaultForeground.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PrefixWmAndIconName</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Whether to prefix the window and icon name with &quot;olmh: &quot;. Default is true.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PrintCommand</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>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 &gt;/dev/null 2&gt;/dev/null’.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ReplyInsertFilter</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>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 &quot;&quot;&quot;$0&quot;&quot;}' or mh directory&gt;/lib/mhl -form mhl.body.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ReverseReadOrder</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SendBreakWidth</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SendWidth</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SkipCopied</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Whether to skip over messages marked for copying when using “View Next Message” and “View Previous Message”. Default is true.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SkipDeleted</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Whether to skip over messages marked for deletion when using “View Next Message” and “View Previous Message”. Default is true.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SkipMoved</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Whether to skip over messages marked for moving to other folders when using “View Next Message” and “View Previous Message”. Default is true.</td>
</tr>
</tbody>
</table>
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 pop up 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 mh 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

```
~/.mh_profile - MH profile
/usr/local/mh6 - MH commands
~/[Mail]/<folder>/.olmhcache - scan folder
~/[Mail]/<folder>/.mh_sequences - sequence definitions
/tmp - temporary files
```

modified 24 March 1994
SEE ALSO

xrdb(1), X Toolkit Intrinsics, OLIT Widget Set, mh(1), enscript(1)

BUGS

- 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 proc does 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 MH 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 MH 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 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.

modified 24 March 1994
NAME
olwm – OPEN LOOK window manager for OpenWindows

SYNOPSIS
olwm [ options ]

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
Most command-line options have counterparts in the resource database. A command-line option will override any setting from the resource database.

-2d Use two-dimensional look. This is the default for monochrome systems.
-3d Use three-dimensional look. This is the default for color systems. This option is ignored for monochrome systems.

-bd color, -bordercolor color
Specifies the border color. See the description of the BorderColor resource.

-bg color, -background color
Specifies the background color. See the description of the Background resource.

-c, -click
Use click-to-focus mode. This is the default focus mode.

-depth depth
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.

-display display-string
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.

-dsdm Specify that olwm should provide the Drop Site Database Management (DSDM) service. This is the default.

-f, -follow
Use focus-follows-mouse mode. Default mode is click-to-focus.

-fn font-name, -font font-name
Set the font for window titles.

-fg color, -foreground color
Specifies the foreground color. See the description of the ForegroundColor resource.

-multi Manage windows on all screens that a display supports. This is the default.

-name resource-name
Use resource-name to look up resources in the resource database.

-nodsdm

modified 25 Nov 1992
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=$!
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.

**INTERNATIONALIZATION 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
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

modified 25 Nov 1992
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

modified 25 Nov 1992
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.

<table>
<thead>
<tr>
<th>MENU</th>
<th>OPERATION</th>
</tr>
</thead>
</table>
| 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.

<table>
<thead>
<tr>
<th>WORKSPACE</th>
<th>MENU</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.)</td>
<td></td>
</tr>
</tbody>
</table>

**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.
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.

WINDOW MENU

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 redispalyes 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.
You can customize \texttt{olwm}'s Workspace Menu by putting a menu description into a file that \texttt{olwm} will read. When it starts up, \texttt{olwm} will first look for a file named by the OLWMENU environment variable. If this variable does not exist, or if the file is not readable, \texttt{olwm} will then look in the file named \texttt{.openwin-menu} in your home directory. If this file is not present or is unreadable, \texttt{olwm} will fall back on the system default menu file. If, for some reason, the system default menu file cannot be found, \texttt{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 \texttt{.openwin-menu.japanese} will take precedence over the file \texttt{.openwin-menu}. \texttt{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 \texttt{AutoReReadMenuFile} resource.

If \texttt{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. \texttt{Olwm} will then attempt to read the next file in the sequence as described above.

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 \texttt{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

modified 25 Nov 1992
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
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
ingstalled, 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 opera-
tions 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 data-
base of all possible locations on the screen where an object may be dropped. These loca-
tions 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 ser-
vise 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 pro-
viding 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 pro-
cess; rather, they control whether olwm provides the DSDM service itself. It is not neces-
sary 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 argu-
ment. 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 lap-
ning 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, oIwm 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: 

\texttt{-sun-open\_look\_glyph=-*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*\_120--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*--*).

\textbf{IconFlashCount} (integer)

Number of times to flash the open/close "zoom" lines. Default value: 3.

\textbf{IconFlashOffTime} (integer)

Amount of time to pause, in microseconds, while open/close "zoom" lines are not visible. Default value: 1.

\textbf{IconFlashOnTime} (integer)

Amount of time to pause, in microseconds, while open/close "zoom" lines are visible. Default value: 20000.

\textbf{IconFont} (font name)

Font used for icon names. Default: lucidasans-12.

\textbf{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.

\textbf{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.

\textbf{KeepTransientsAbove} (boolean)

Specifies whether \texttt{olwm} should attempt to keep transient windows above their owner window. Default value: false.

\textbf{KeyboardCommands} (enumeration) *

Permissible values for this resource are \texttt{SunView1}, \texttt{Basic}, and \texttt{Full}. Values are case-sensitive. In \texttt{Full} mode, all OPEN LOOK Mouseless commands implemented by the window manager are active. See the section on Mouseless Navigation for further information. In \texttt{Basic} mode, the keys active are Open, Front, Help, and the colormap keys. In \texttt{SunView1} mode, the only keys active are Open and Front. Default value: \texttt{Basic}.

\textbf{MenuAccelerators} (boolean)

Determines whether menu accelerators are active. Used in conjunction with the \textbf{WindowMenuAccelerators} resource. Both must be set to true for menu accelerators to be active. Default value: true.

\textbf{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: calc tool 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.

modified 25 Nov 1992
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* 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 which saves the currently running clients into the OpenWindows startup script $HOME/.openwin-init. *Default value:*
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

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 may cause clashes with XView clients which only use the global setting.
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.

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

```
```

to the .Xdefaults file.

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:
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:

```plaintext
olwm.KeyboardCommand.Stop
```

and the resource specification for setting the Back menu accelerator function would be:

```plaintext
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 (up-arrow)  
Move up one item.

Down (down-arrow)  
Move down one item.

Left (left-arrow)  
Move left one item.

Right (right-arrow)  
Move right one item.

JumpUp (Control up-arrow)  
Move up ten items.

JumpDown (Control down-arrow)  
Move down ten items.

JumpLeft (Control left-arrow)  
Move left ten items.

JumpRight (Control right-arrow)  
Move right ten items.

RowStart (Home, R7)  
Move to the start of the current row.

RowEnd (End, R13)  
Move to the end of the current row.

DataStart (Control-Home)  
Move to the start of the data.

DataEnd (Control-End)  
Move to the end of the data.

FirstControl (Control-[)  
Move to the first item.

LastControl (Control-])  
Move to the last item.

NextElement (Tab, Control-Tab)  
Move to the next item.

PreviousElement (Shift-Tab, Control-Shift-Tab)  
Move to the previous item.

Open (Alt-L7) *  
Open the object with the input focus.

Help (Help) *  
Bring up Magnify Help on the object under the pointer.

LockColormap (Control-L2) *  
Install the colormap of the subwindow under the pointer, and give the colormap focus to the top-level window containing the pointer. See Colormap Installation for further details.
UnlockColormap (Control-L4) *
  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 start-
  ing 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 win-
  dow 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.

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 pathname 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.
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 [ options ]

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  
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.

In addition, openwin accepts the following command line arguments that the server does not accept:

[-server Xsun]  
This tells openwin which server binary to start. The default is $OPENWINHOME/bin/Xsun.

[-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. 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. See the OpenWindows documentation for more information.

[-auth protocol-name]  
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.

[-includedemo]  
This option indicates that the path to the demo directory should be included in the user’s search path (see PATH below).

[-wm wm-command]  
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 wm-command of multiple words must be quoted. This option is likely to be ignored if the user has a $HOME/.xinitrc script of their
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:

```bash
# @(#)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
  $OW_WINDOW_MANAGER & wmpid=$!
  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
```

modified 22 February 1994
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), xdb(1)
NAME
owplaces – list client applications running on a display

SYNOPSIS
owplaces [ −display displaystring ] [ −timeout nseconds ]
   [ −single | −multi | −pointer ]
   [ −all | −local | −remote | −host hostname ]
   [ −script ] [ −output filename ] [ −ampersand ] [ −tw ]
   [ −silent ]

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
The basic options are as follows:
−display string
   Specify the name of the display to manage. Overrides the DISPLAY
   environment variable, if any.
−timeout nsecs
   Specifies the time to wait for a reply for those clients that are updating
   their command lines. Default is 30 seconds.

MODE OPTIONS
Options that control which clients are printed:
−single
   Search only the default screen of the specified display for clients.
−multi
   Search all screens of the specified display for clients. This is the default.
−pointer
   Print only the client selected by clicking the mouse in the desired win-
   dow.

CLIENT MACHINE FILTER OPTIONS
Options to filter clients based on host machine (as specified by the
WM_CLIENT_MACHINE property):
−all
   Print clients on all hosts. This is the default.
−local
   Print clients local to this machine only.
−remote
   Print clients that are remote to this machine.
−host hostname
   Print clients from hostname only.

INITIALIZATION SCRIPT OPTIONS
Options useful to producing shell scripts:
−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.
−output filename
   Directs output to filename instead of stdout. If filename exists, it is
   renamed as filename.BAK.
−ampersand
   Appends an ampersand character (&) to end of each command line.
   This is implied if −script is used.
−tw
   Prepends the command toolwait(1) to each client. The −ampersand
   option has no affect if this option is specified.
owplaces (1) User Commands OpenWindows Desktop 3.5

-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)
<table>
<thead>
<tr>
<th>NAME</th>
<th>pageview – PostScript language previewer for OpenWindows</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>pageview [-mono] [-w paperwidth] [-h paperheight] [-dpi dots/inch]</td>
</tr>
<tr>
<td></td>
<td>[ -mcd colordensity ] [ -page pagenum</td>
</tr>
<tr>
<td></td>
<td>[ -dir directory ]</td>
</tr>
<tr>
<td></td>
<td>[ -left</td>
</tr>
<tr>
<td></td>
<td>[ -low_memory ] [ -usage ] [ -v ] [ -verbose ] [ psfile ]</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>pageview is an interactive PostScript previewer. pageview renders a document, a page</td>
</tr>
<tr>
<td></td>
<td>at a time, onto an offscreen bitmap of arbitrary size, resolution and orientation. You</td>
</tr>
<tr>
<td></td>
<td>can then adjust the viewing window’s size to see as much of the page as desired. The</td>
</tr>
<tr>
<td></td>
<td>mouse buttons are used to position the page under the window in two different modes.</td>
</tr>
<tr>
<td></td>
<td>The left button moves the page in &quot;relative mode&quot;. This allows you to move the page in a</td>
</tr>
<tr>
<td></td>
<td>physically intuitive way. You press the left button on the page and while you drag the</td>
</tr>
<tr>
<td></td>
<td>mouse around, the spot on the page that was under the mouse cursor when you pressed</td>
</tr>
<tr>
<td></td>
<td>the button remains stationary relative to the cursor.</td>
</tr>
<tr>
<td></td>
<td>The middle button moves the page in &quot;absolute mode&quot;. This allows you to easily get to</td>
</tr>
<tr>
<td></td>
<td>the edges of the document, especially when the DPI is large and/or the window is small.</td>
</tr>
<tr>
<td></td>
<td>When you press the middle button on a point in the window, the page is adjusted so that</td>
</tr>
<tr>
<td></td>
<td>the same relative point on the page is under the mouse. For example, if you press the</td>
</tr>
<tr>
<td></td>
<td>middle button at the top right corner of the window, you will see the top right corner of</td>
</tr>
<tr>
<td></td>
<td>the page. A little experimentation with a page at 300 dpi and you will find this mode</td>
</tr>
<tr>
<td></td>
<td>indispensable.</td>
</tr>
<tr>
<td></td>
<td>The three menu buttons across the top of the main window are described below:</td>
</tr>
<tr>
<td>File</td>
<td>The File menu is used to bring up the Load... or Print... dialogs.</td>
</tr>
<tr>
<td>Load...</td>
<td>brings up a dialog which prompts for a directory and filename to load a new PostScript file.</td>
</tr>
<tr>
<td>Print...</td>
<td>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.</td>
</tr>
<tr>
<td>View</td>
<td>The View menu allows you to move to the Next, Previous, First and Last pages of multipage documents.</td>
</tr>
<tr>
<td>Edit</td>
<td>The edit menu has two choices, PostScript and Properties.</td>
</tr>
<tr>
<td>PostScript...</td>
<td>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.</td>
</tr>
<tr>
<td>Properties...</td>
<td>brings up the property sheet for page size, orientation and resolution.</td>
</tr>
<tr>
<td>DPI:</td>
<td>This lets you change the “resolution&quot; 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</td>
</tr>
</tbody>
</table>
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.
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.

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.

Sets the page number of the document that is to be displayed when pageview starts.

Sets the current working directory to directory so that you can type filenames at the "Load File: " prompt relative to directory.

Sets the rotation of the page.

Sets the job timeout value. The default job timeout value is 30 seconds.

Turns on antialiasing.

Runs a low memory version of pageview. This version does not create the off-screen 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.

Prints out all valid command line options.

Prints out the current version of pageview.

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

Ip(1)
OpenWindows user documentation

modified 26 February 1992
NAME  
perfmeter – display system performance values in multiple dials or strip charts

SYNOPSIS  
perfmeter [-a] [-d] [-g] [-h h-hand-int] [-i] [-m m-hand-int] [-n sample-filename]
[-V] [-Wn] [+Wn] [hostname]

AVAILABILITY  
This command is available with the OpenWindows environment.

DESCRIPTION  
perfmeter is an OpenWindows XView utility that displays performance values (statistics) for a given hostname. If no host is specified, statistics on the current host are metered. The rstatd(1M) daemon must be running on the machine being metered.

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 sample-time 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.

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.

OPTIONS  
-a  Display all the performance values simultaneously.
-d  Start the perfmeter display, by showing metered dials.
-g  Start the perfmeter display, by showing strip charts.
-h h-hand-int  Set the hour-hand interval to h-hand-int seconds.
-i  Log the perfmeter samples to a file.
-m m-hand-int  Set the minute hand interval to m-hand-int seconds.
-n sample-filename  Name of the file to use, when saving perfmeter samples.
-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.
-p page-length  Sets the page length for perfmeter samples written to file. The default is 60 lines per page.
-s sample-time  Set the sample time to sample-time seconds.

modified 03 December 1993
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.

Show version number

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.

Multiple dials or strip charts will be displayed initially in a horizontal direction.

If the `-g` option is present, then initially display the strip charts in line format.

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.

If the `-g` option is present, then initially display the strip charts in a solid filled format.

Multiple dials or strip charts will be displayed initially in a vertical direction.

Start `perfmeter` with no title line.

Start `perfmeter` with a title line present.

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.

modified 03 December 1993
<table>
<thead>
<tr>
<th>Resource:</th>
<th>Value:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deskset.perfmeter.intrMaxValues</td>
<td>Starting, lowest and highest maximum values (numeric)</td>
<td>The starting, lowest and highest maximum values for the intr graph or dial.</td>
</tr>
<tr>
<td>deskset.perfmeter.diskMaxValues</td>
<td>Starting, lowest and highest maximum values (numeric)</td>
<td>The starting, lowest and highest maximum values for the disk graph or dial.</td>
</tr>
<tr>
<td>deskset.perfmeter.cntxtMaxValues</td>
<td>Starting, lowest and highest maximum values (numeric)</td>
<td>The starting, lowest and highest maximum values for the cntxt graph or dial.</td>
</tr>
<tr>
<td>deskset.perfmeter.loadMaxValues</td>
<td>Starting, lowest and highest maximum values (numeric)</td>
<td>The starting, lowest and highest maximum values for the load graph or dial.</td>
</tr>
<tr>
<td>deskset.perfmeter.collMaxValues</td>
<td>Starting, lowest and highest maximum values (numeric)</td>
<td>The starting, lowest and highest maximum values for the coll graph or dial.</td>
</tr>
<tr>
<td>deskset.perfmeter.errMaxValues</td>
<td>Starting, lowest and highest maximum values (numeric)</td>
<td>The starting, lowest and highest maximum values for the err graph or dial.</td>
</tr>
<tr>
<td>deskset.perfmeter.collectWhenObscured</td>
<td>True, False (True)</td>
<td>Indicates whether data will be monitored and cached, when the perfmeter window is fully obscured.</td>
</tr>
<tr>
<td>deskset.perfmeter.hasTitle</td>
<td>True, False (False)</td>
<td>Indicates whether the perfmeter window has a title line.</td>
</tr>
<tr>
<td>deskset.perfmeter.labelFont</td>
<td>Font name string</td>
<td>The name of the font used to display all the labels for the dials and graphs.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.saveSamples</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Values:</td>
<td>True, False (False)</td>
<td></td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.sampleFilename</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.samplePageLength</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.cpuColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.pktsColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.pageColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.swapColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.intrColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.diskColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.cntxtColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.loadColor</td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

modified 03 December 1993
<table>
<thead>
<tr>
<th>Description</th>
<th>The color of the load strip chart.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.collColor</td>
</tr>
<tr>
<td>Values:</td>
<td>Color name string or hexadecimal color specification string</td>
</tr>
<tr>
<td>Description:</td>
<td>The color of the coll strip chart.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.errColor</td>
</tr>
<tr>
<td>Values:</td>
<td>Color name string or hexadecimal color specification string</td>
</tr>
<tr>
<td>Description:</td>
<td>The color of the err strip chart.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.ceilingColor</td>
</tr>
<tr>
<td>Values:</td>
<td>Color name string or hexadecimal color specification string</td>
</tr>
<tr>
<td>Description:</td>
<td>The color used when any strip chart goes above its ceiling value.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.ceilingStyleSolid</td>
</tr>
<tr>
<td>Values:</td>
<td>True, False (False)</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.cpuCeiling</td>
</tr>
<tr>
<td>Values:</td>
<td>cpu ceiling value (numeric)</td>
</tr>
<tr>
<td>Description:</td>
<td>The ceiling value to use when displaying the cpu strip chart. Values above this will be displayed in a different color on color screens.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.pktsCeiling</td>
</tr>
<tr>
<td>Values:</td>
<td>pkts ceiling value (numeric)</td>
</tr>
<tr>
<td>Description:</td>
<td>The ceiling value to use when displaying the pkts strip chart. Values above this will be displayed in a different color on color screens.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.pageCeiling</td>
</tr>
<tr>
<td>Values:</td>
<td>page ceiling value (numeric)</td>
</tr>
<tr>
<td>Description:</td>
<td>The ceiling value to use when displaying the page strip chart. Values above this will be displayed in a different color on color screens.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.swapCeiling</td>
</tr>
<tr>
<td>Values:</td>
<td>swap ceiling value (numeric)</td>
</tr>
<tr>
<td>Description:</td>
<td>The ceiling value to use when displaying the swap strip chart. Values above this will be displayed in a different color on color screens.</td>
</tr>
<tr>
<td>Resource:</td>
<td>deskset.perfmeter.intrCeiling</td>
</tr>
<tr>
<td>Values:</td>
<td>intr ceiling value (numeric)</td>
</tr>
<tr>
<td>Description:</td>
<td>The ceiling value to use when displaying the intr strip chart. Values</td>
</tr>
</tbody>
</table>

modified 03 December 1993
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**
printtool – OpenWindows tool for printing files

**SYNOPSIS**
printtool [-P printer] [-v] [-?]

**DESCRIPTION**
Printtool is a DeskSet utility that provides a Graphical User Interface (GUI) for the `lp(1)`, `cancel(1)`, and `lpstat(1)` printing functions. **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.

**OPTIONS**
In addition to the generic tool arguments supported by `xview(7)`, **printtool** accepts the following options:

- `-P printer` Send output to the named printer.
- `-v` Show the version number and the usage message of this release of the **printtool** program.
- `-?` Same as `-v`

**USAGE**
**Printtool** is part of the OpenWindows user environment. For more information on the basic functions of OpenWindows, see the Solaris User’s Guide

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.

Users may also drag and drop file icons from `filemgr(1)` or mail attachments from `mailtool(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.

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.

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:

modified 19 November 1993
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:

<table>
<thead>
<tr>
<th>Resource:</th>
<th>Values:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deskset.printtool.checkInterval</td>
<td>Status check interval (numeric)</td>
<td>This is the number of seconds between status checks on the printer. The default value is 10 seconds.</td>
</tr>
<tr>
<td>deskset.printtool.headerPage</td>
<td>true, false</td>
<td>This is the flag for printing with header page or not. The default value is false (no header page).</td>
</tr>
<tr>
<td>deskset.printtool.printMethodOverride</td>
<td>true, false</td>
<td>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).</td>
</tr>
<tr>
<td>deskset.printtool.lastPrintMethod</td>
<td>Print method string or full path of script string</td>
<td>This is the print method for printing instead of the default method if deskset.printtool.printMethodOverride is set to true.</td>
</tr>
<tr>
<td>deskset.printtool.notifyFlash</td>
<td>true, false</td>
<td>If set to true, printtool will flash the window or icon when all print jobs are done. The default value is false (no flash).</td>
</tr>
<tr>
<td>deskset.printtool.notifyBeep</td>
<td>true, false</td>
<td>If set to true, printtool will beep when all print jobs are done. The default value is false (no beep).</td>
</tr>
</tbody>
</table>

modified 19 November 1993
SEE ALSO accept(1M), cancel(1), enable(1), filemgr(1), lp(1), lpadmin(1), lpfiler(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
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.

Upon window system startup, the .OWdefaults file is merged into the X resource manager as specified in the openwin-sys 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.

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.

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.

BasicLocale (locale name)
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.

Beep (enumeration)
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
**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: Other Options, Key Repeat

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-normal-120-normal-normal-normal
Properties category: Fonts
Category control: Typeface

modified 23 February 1994
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-120-120-120-120-120`

**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 screensaver 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

---

modified 23 February 1994
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 Equivalents

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.

modified 23 February 1994
WorkspaceBitmapFg (color)
Specifies the color for the foreground of the workspace bitmap which \texttt{olwm} tiles on the root window if the \texttt{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, \texttt{olwm} will set the root window's background color to the color specified by this resource if the \texttt{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 \texttt{olwm} decorates the workspace (root window). Permissible values are "paintcolor" and "tilebitmap". If the value is "paintcolor", \texttt{olwm} paints the workspace with the color indicated by the \texttt{WorkspaceColor} resource. If the value is "tilebitmap", \texttt{olwm} tiles the bitmap indicated by the \texttt{WorkspaceBitmapFile} resource using the colors specified by the \texttt{WorkspaceBitmapFg} and \texttt{WorkspaceBitmapBg} resources.
Default value: paintcolor
Properties category: Color
Category control: Pattern
The default specifies to use the color selected by WorkspaceColor.

OPTIONS
The \texttt{props} program accepts all standard command line options accepted by the OLIT toolkit. In addition, it accepts the following.

\texttt{--init}
This is the way \texttt{props} is started by \texttt{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 \texttt{.OWdefaults} file in the user's home directory: key click, key repeat, beep duration, screen saver activation and timeout. When \texttt{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
\texttt{$HOME/.OWdefaults}
Contains the resource name-value pairs written by the \texttt{props} program.

\texttt{$HOME/openwin-menu-programs}
Contains the user's custom programs menu.

\texttt{$HOME/openwin-menu-programs.BAK}

modified 23 February 1994
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 rash – Sun Raster to PostScript translator

SYNOPSIS rash [-2] [ -nxm ] [ -s width height ] [ [ -H height ] [ -S height ] [ -W width ]
                   [ -l xpos ypos ] [ -c xpos ypos ] [ -R angle ] [ -e ] [ -i ] [ -m ] [ -n ] [ -p prolog ]
                   [ -r ] [ -w wrapper ] [ filename... ]

DESCRIPTION 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.

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.

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.

rash is most often used with the PreLimm filter manager, part of the NeWSprint package.

OPTIONS

-2 Scale the image by a factor of 2. Each pixel in the raster file is printed as 4 pixels (2 x 2) on the output device.

-nxm Scale the image by a factor of n in the horizontal direction and m in the vertical direction. If only one number is specified, both dimensions are scaled evenly.

-s width height Scale the image to the specified width and height. This option does not preserve the original raster’s aspect ratio.

-H height Scale the image (preserving aspect ratio) to the specified height.

-S height Same as -H.

-W width Scale the image (preserving aspect ratio) to the specified width.

-l xpos ypos Locate the lower left corner of the image xpos and ypos units in the x and y direction. The origin is at the lower left corner of the page.

-c xpos ypos Locate the center of the image xpos and ypos units from the center of the page.

-R angle Rotate the image to the specified angle, in degrees. (Probably most useful with the -c option.)

-e Output Encapsulated PostScript (EPSF). The -c, -2, -nxm, 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.

-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.

-m If the input is a color raster file, convert it to 8-bit gray scale.

modified 9 Nov 1990
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.

`prolog` Output the PostScript to the specified file rather than to the standard output.

Print in landscape mode

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*

*PreLimn 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 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

−fnn The number of fish. The default is 2. This option is ignored if you have used the fish properties sheet and a ~/.fishrc file can be found.

−bn The number of bubbles. The suggested number is 0 since they take a significant amount of CPU power. The default is 0.

−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 ~/.fishrc file can be found.

−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 ~/.fishrc file can be found.

−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 ~/.fishrc file can be found.

−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.

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 ~/.fishrc file. realxfishdb attempts to read this file when it starts.

ENVIRONMENT

There are image files in the $OPENWINHOME/share/images/fish directory which are required to run realxfishdb.

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 $OPENWINHOME directory the final path attempted is the ./fish directory relative to where the executable lives.

modified 06 March 1991
### 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  rpc.cmsd – calendar manager service daemon

SYNOPSIS  /usr/openwin/bin/rpc.cmsd [-d] [-s]

DESCRIPTION  rpc.cmsd is a small database manager for appointment and resource-scheduling data. Its 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 received.

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
- `/usr/spool/calendar/callog.$USER`
- `/etc/inetd.conf`

modified 9 November 1993
NAME  
shelltool – run a shell (or other program) in an OpenWindows terminal window

SYNOPSIS  
shelltool [ −C ] [ −B boldstyle ] [ −I command ] [ generic-tool-arguments ]
[ program [ arguments ] ]

DESCRIPTION  
shelltool is a standard OpenWindows facility for shells or other programs that use a standard tty-based interface.

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 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.

OPTIONS  
−C  Redirect system console output to this shelltool.
−B boldstyle  Set the style for displaying bold text to boldstyle. boldstyle 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.
−I command  Pass command to the shell. SPACE characters within the command must be escaped.

generic-tool-arguments  
shelltool accepts the generic tool arguments listed in xview(7).

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  
The command format for this file is:

#  Comment.
set variable  Turn on the specified variable.
mapi key text  When key is typed pretend text was input.
mapo key text  When key is typed pretend text was output.

The only currently defined variable is pagemode. key is one of L1-L15, F1-F15, T1-T15, R1-R15, LEFT, or RIGHT on SPARC keyboards. On x86 keyboards key is one of F1-F12, End, PgDn, PgUp, HOME, LEFT, or RIGHT (see note below). text may contain escapes such

modified 12 May 1994
as \E, \n, `X, etc. (ESC, RETURN, and CTRL-X, respectively). See curs_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 XVView. 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.

\E[1t – open
\E[2t – close (become iconic)
\E[3t – move, with interactive feedback
\E[3;TOP;LEFTt – move, to TOP LEFT (pixel coordinates)
\E[4t – stretch, with interactive feedback
\E[4;HT;WIDTHt – stretch, to HT WIDTH size (in pixels)
\E[5t – front
\E[6t – back
\E[7t – refresh
\E[8;ROWS;COLSt – stretch, to ROWS COLS size (in characters)
\E[11t – report if open or iconic by sending \E[1t or \E[2t
\E[13t – report position by sending \E[3;TOP;LEFTt
\E[14t – report size in pixels by sending \E[4;HT;WIDTHt
\E[18t – report size in characters by sending \E[8;ROWS;COLSt
\E[20t – report icon label by sending \E|label|\E\n\E[21t – report tool header by sending \E|header|\E\n\E|text|\E\n – set tool header to text
\E|file|\E\n – set icon to the icon contained in file; file must be in iconedit output format
\E|label|\E\n – set icon label to label
\E>[OPT;…h – turn SB OPT on (OPT = 1 => pagemode), for example, \E>[1;3;4h
\E>[OPT;…k – report OPT; sends \E>[OPT1 or \E>[OPTh for each OPT
\E>[OPT;…l – turn OPT off (OPT = 1 => pagemode), for .B \E>[1;3;

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 "\[Elnl

# dynamically set the label on the icon:
alias iheader 'echo -n "\[El"

# dynamically set the image on the icon:
alias icon 'echo -n "\[EIl"
```

FILES
~/.ttyswrc
/usr/openwin/lib/.ttyswrc
/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

modified 12 May 1994
NAME  snapshot – capture some or all of a screen image and save to a raster file

SYNOPSIS  snapshot [ −d default directory ] [ −f default filename ] [ −g ] [ −l filename ] [ −n ] [ −v ]

AVAILABILITY  This command is available with the OpenWindows environment.

DESCRIPTION  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.

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.

Images can be saved to disk in Sun rasterfile format, or printed on a PostScript printer. There are various printer options (see below).

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).

OPTIONS  
−d default-directory  
Name of the default directory to use for load and save operations.

−f default-filename  
Name of the default filename to use for load and save operations.

−g  Display color images using a greyscale ramp.

−l filename  
Name of the file to automatically load and view on startup.

−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.

−v  Print the version number of this release of the snapshot program.

USAGE  snapshot operates using a combination of control panels with various buttons, choices and text fields to define the actions performed and the files used.

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.

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.
Print

This button displays a menu with two options:

Print Snap
Sends the currently snapped or loaded image to the printer with the current options.

Options
The options displayed in the Print Options window follow.

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.

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.

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.

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.

Double Size:
Specifies whether the snapshot is to be doubled in width and height.

Monochrome Printer
Select this option if the printed output is to be in monochrome.

Snap Type:
The type of image to snap; a window, a rectangular region of the screen, or the entire screen.

Snap Delay:
Delays the snapping of the picture for the specified number of seconds.

Beep During Countdown
Indicates if a beep should be sounded every second as the snap delay is counted down.

Hide Window During Capture
The snapshot window is made invisible before the snapshot is taken. The timer delay is forced to eight seconds.

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

modified 9 March 1992
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 filename ] [ −toolkitoption... ]

DESCRIPTION  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.

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.

OPTIONS  −save_file filename
          Start up using the specified save file.

RESOURCES  spider understands all of the core X Toolkit resource names and classes as well as:

bell (class Bell)
          Boolean which specifies whether spider will use the bell to when complaining about illegal moves. The default is "true".

confirm (class Confirm)
          Boolean which specifies whether spider will use ask for confirmation before certain actions. The default is "true".

replayTime (class ReplayTime)
          Specifies the time (in microseconds) to pause between each move when showing a Replay. The default is 200.

roundsCards (class RoundCards)
          Specifies whether to use rounded cards are not. Rounded cards look better, but are considerably slower. The default is "on".

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 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.

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".

helpDir (class HelpDir)
          Specifies where to look for the help files.

modified 30 Jan 1990
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.

modified 30 Jan 1990
NAME  
tapetool – OpenWindows tool for reading files from tape or archiving files to tape

SYNOPSIS  
tapetool [ generic-tool-arguments ]

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  
generic-tool-arguments

tapetool accepts the generic tool arguments described in xview(7).

USAGE  
tapetool is part of the OpenWindows user environment.

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’.

‘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’.

There are three ways to read files:

Read Selected
Reads the files selected in the ‘Tape Contents’ window.

Read Entire List
Reads all the files listed in the ‘Tape Contents’ window.

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 pathnames.

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.

Props gives a property sheet for specifying tar(1) options.

Device:
/dev/rmt/nn  SCSI tape drives

Host Name:
Specifies the name of host system where the reading or archiving of
files will be applied. The default is the current host name.

**Tar Options**

**Write:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No SCCS</td>
<td>F option to <strong>tar</strong>(1), exclude all directories named SCCS from tarfile.</td>
</tr>
<tr>
<td>No SCCS+</td>
<td>FF option to <strong>tar</strong>(1), exclude all directories named SCCS, all files with <code>.o</code> as their suffix, and all files named <code>errs</code>, <code>core</code>, and <code>a.out</code>.</td>
</tr>
<tr>
<td>Block I/0</td>
<td>b option to <strong>tar</strong>(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 <strong>tar</strong> when it is run. Larger blocking factors result in better throughput, longer blocks on nine-track tapes, and better media utilization.</td>
</tr>
<tr>
<td>Sym Links</td>
<td>h option to <strong>tar</strong>(1), follow symbolic links as if they were normal files or directories. Normally, <strong>tar</strong> does not follow symbolic links.</td>
</tr>
<tr>
<td>Show Errs</td>
<td>I option to <strong>tar</strong>(1), Display error messages if all links to archived files cannot be resolved. If not used, no error messages are printed.</td>
</tr>
<tr>
<td>Suppress</td>
<td>o option to <strong>tar</strong>(1), Suppress information specifying owner and modes of directories that <strong>tar</strong> normally places in the archive.</td>
</tr>
</tbody>
</table>

**Read:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Check</td>
<td>i option to <strong>tar</strong>(1), Ignore directory checksum errors.</td>
</tr>
<tr>
<td>Mod Time</td>
<td>m option to <strong>tar</strong>(1), Do not extract modification times of extracted files. The modification time will be the time of extraction.</td>
</tr>
<tr>
<td>Orig Mode</td>
<td>p option to <strong>tar</strong>(1), Restore the named files to their original modes, ignoring the present <strong>umask</strong>(2). SetUID and sticky information are also extracted if you are the super-user.</td>
</tr>
</tbody>
</table>

**Delete Dir:**

For writing files to tape

- ‘NONE’      retains filename path exactly as specified.
- ‘ALL’       Extracts the entire path from `filename` and writes just the `filename`.  
- ‘PATTERN’   extracts the `pattern` specified and writes the `filename` left over.

**Other:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Err Exit</td>
<td>e option to <strong>tar</strong>(1), If any unexpected errors</td>
</tr>
</tbody>
</table>
occur \texttt{tar} aborts immediately with a positive exit status.

\textbf{Exclude} \hspace{2cm} \textbf{X} option to \texttt{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.

\textbf{Apply} \hspace{2cm} Menu button that applies the options that have been set.
\textbf{Reset} \hspace{2cm} Menu button that resets the defaults.

You may drag and drop files from \texttt{filemgr}(1) onto the \texttt{tapetool} write window for writing files to tape. You can also drop files onto the \texttt{tapetool} icon for writing.

\begin{verbatim}
FILES /dev/rmt/ SCSI tape interface
BUGS See bugs for \texttt{tar}(1)
SEE ALSO \texttt{tar}(1), \texttt{filemgr}(1)
\end{verbatim}

OpenWindows user documentation

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

\textit{modified 9 March 1992}
**NAME**
textedit – XView-based text editor with mouse support

**SYNOPSIS**
textedit [ generic-tool-arguments ] [ –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 ] [ filename ]

**DESCRIPTION**
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.
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.

**OPTIONS**
generic-tool-arguments
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.

–Ei on | off
–auto_indent
Choose whether or not to automatically indent newly-opened lines. The default
is off. Corresponds to the auto_indent Text default.

–Eo on | off
–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 mes-
statement. Corresponds to Store_self_is_save.

–Er on | off
–read_only
Turn read-only mode on or off. When on, text cannot be modified.

–Ec N
–checkpoint count
Checkpoint after every count editing operation. If count 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.

–EL lines
–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.
textedit (1) User Commands OpenWindows Desktop 3.5

−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 tabs
  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

modified 06 December 1993
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

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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-Primary</td>
<td>META-X</td>
<td>Erases the primary selection and moves it to the Clipboard.</td>
</tr>
<tr>
<td>Find-Primary</td>
<td>META-F</td>
<td>Searches the text for the pattern specified by the primary selection or by the Clipboard, if there is no primary selection.</td>
</tr>
<tr>
<td>Copy-to-Clipboard</td>
<td>META-C</td>
<td>Copies the primary selection to the Clipboard.</td>
</tr>
<tr>
<td>Paste-Clipboard</td>
<td>META-V</td>
<td>Inserts the Clipboard contents at the insertion point.</td>
</tr>
<tr>
<td>Copy-then-Paste</td>
<td>META-P</td>
<td>Copies the primary selection to the insertion point (through the Clipboard).</td>
</tr>
<tr>
<td>Go-to-EOF</td>
<td>CTRL-RETURN</td>
<td>Moves the insertion point to the end of the text and positions the text so the insertion point is visible.</td>
</tr>
</tbody>
</table>

Keyboard Functions

The commands indicated by use of the function keys are:

<table>
<thead>
<tr>
<th>Command</th>
<th>SPARC</th>
<th>x86</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>L1</td>
<td>Esc</td>
<td>Aborts the current command.</td>
</tr>
<tr>
<td>Again</td>
<td>L2</td>
<td>Meta-a</td>
<td>Repeats the previous editing sequence since a primary selection was made.</td>
</tr>
<tr>
<td>Undo</td>
<td>L4</td>
<td>Meta-z</td>
<td>Undoes a prior editing sequence.</td>
</tr>
<tr>
<td>Front</td>
<td>L5</td>
<td>Meta-r</td>
<td>Makes the window completely visible (or hides it, if it is already exposed).</td>
</tr>
<tr>
<td>Copy</td>
<td>L6</td>
<td>Meta-c</td>
<td>Copies the primary selection, either to the Clipboard or at the closest end of the secondary selection.</td>
</tr>
<tr>
<td>Open</td>
<td>L7</td>
<td>Meta-w</td>
<td>Makes the window iconic (or normal, if it is already iconic).</td>
</tr>
<tr>
<td>Paste</td>
<td>L8</td>
<td>Meta-v</td>
<td>Copies either the secondary selection or the Clipboard at the insertion point.</td>
</tr>
<tr>
<td>Find</td>
<td>L9</td>
<td>Meta-f</td>
<td>Searches for the pattern specified by, in order, the secondary selection, the primary selection, or the Clipboard.</td>
</tr>
<tr>
<td>Cut</td>
<td>L10</td>
<td>Meta-x</td>
<td>Erases either the primary or the secondary selection and moves it to the Clipboard.</td>
</tr>
<tr>
<td>Help</td>
<td>F1</td>
<td>F1</td>
<td>Produces help text.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File</strong></td>
<td>A pull-right menu item for file operations.</td>
</tr>
<tr>
<td><strong>Edit</strong></td>
<td>A pull-right menu item equivalent of the editing function keys. The <strong>Edit</strong> submenu provides <strong>Again</strong>, <strong>Undo</strong>, <strong>Copy</strong>, <strong>Paste</strong>, and <strong>Cut</strong> (same as function keys L2, L4, L6, L8, and L10).</td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>A pull-right menu item for controlling the way text is displayed and line display format.</td>
</tr>
<tr>
<td><strong>Find</strong></td>
<td>A pull-right menu item for find and delimiter matching operations.</td>
</tr>
<tr>
<td><strong>Extras</strong></td>
<td>A user definable pull-right menu item. The <strong>Extras</strong> standard submenu is controlled by <code>$OPENWINHOME/lib/locale/&lt;locale&gt;/xview/.text_extras_menu</code> This file has the same syntax as <code>.openwin-menu</code> file. See the <code>xview</code> 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”.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>//.textswrc</code></td>
<td>specifies bindings of filters to function keys</td>
</tr>
<tr>
<td><code>$OPENWINHOME/lib/locale/&lt;locale&gt;/xview/.text_extras_menu</code></td>
<td>specifies bindings of filters for the extras menu pull-right items</td>
</tr>
<tr>
<td><code>$OPENWINHOME/share/src/xview/examples/textsw/textswrc</code></td>
<td>contains useful filters, including <code>shift_lines</code> and <code>capitalize</code>.</td>
</tr>
<tr>
<td><code>filename%</code></td>
<td>contains <code>filename</code> is available here after a <strong>Save Current File</strong> menu operation</td>
</tr>
<tr>
<td><code>textedit.pid</code></td>
<td>edited version of <code>filename</code>; generated in response to fatal internal errors</td>
</tr>
<tr>
<td><code>/tmp/Text*</code></td>
<td>editing session logs</td>
</tr>
</tbody>
</table>

### 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
align_equals
capitalize [-u | -l | -c ]
insert_brackets l r
remove_brackets l r
shift_lines [-t ] n

DESCRIPTION
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 .textswrc file or adding them to the text 'Extras' menu in your .text_extras_menu 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.

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:

big_long_expression = x;
shorter_expr = y;
z+ = 1;

becomes:

big_long_expression = x;
shorter_expr = y;
z += 1;

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.

insert_brackets surrounds the selection (or the standard input) with the specified character sequences. l and r are the left- and right-bracketing characters, respectively.

remove_brackets removes the left- and right-bracketing characters, specified by l and r, respectively from the selection (or the standard input).

shift_lines adjusts indentation of the selection (or the standard input) by n spaces, and replaces the selection with the adjusted text (or writes to the standard output).

shift_lines adjusts to the left when n is negative. If -t is specified, the lines are shifted left or right by n tab stops. The default is 8 spaces per tab stop, but if the first line of the
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  toolwait – control client program startup

SYNOPSIS  toolwait [ −display displaystring ] [ −timeout nseconds ] [ −help ] command

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 string  Specify the name of the display that toolwait should monitor. Overrides the DISPLAY environment variable, if any.

−timeout nsecs  Specifies the time to wait for the client to start up. Toolwait will exit if the client hasn’t started by nsecs 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:

example% toolwait cmdtool -Wp 0 0 -Ws 557 95 -WP 263 833

To start a mailtool on display "host2" with a timeout of 25 seconds, use the following command:

example% toolwait -display host2:0 -timeout 25 mailtool

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.

modified 30 March 1994
NAME
  ttce2xdr – convert ToolTalk Classing Engine type tables to XDR format

SYNOPSIS
  ttce2xdr [-xn] -d user | system
  ttce2xdr [-xn] -d network [OPENWINHOME-from [OPENWINHOME-to]]
  ttce2xdr [-h]
  ttce2xdr [-v]

DESCRIPTION
  ttce2xdr converts ToolTalk types stored in the Classing Engine data base, used by Tool-
  Talk 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 ttsession
  is started for the user, so this command generally only needs to be used by administra-
  tors updating types data bases common to systems or the network.
  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.

OPTIONS
  -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

    The types are written to the XDR data base which is:
    user -- ~/.tt/types.xdr
    system -- /etc/tt/types.xdr
    network -- $OPENWINHOME/etc/tt/types.xdr

    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.

modified 19 January 1994
ENVIRONMENT

CEPATH
In Classing Engine mode, \texttt{tt\_type\_comp} will use this variable for its definition of where the databases are located. See \texttt{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 \texttt{tt\_type\_comp}(1), \texttt{ttsession}(1), \texttt{ce\_db\_build}(1), \texttt{ce\_db\_merge}(1)
NAME
ttcp — copy files and inform the ToolTalk service

SYNOPSIS
ttcp [-pL] filename1 filename2
ttcp -r [-pL] directory1 directory2
ttcp [-prL] filename ... directory
ttcp -h | -v

DESCRIPTION
The ttcp utility invokes the cp(1) utility to copy files and directories, and informs ToolTalk about its actions so that the ToolTalk objects associated with those files and directories can also be copied.

OPTIONS
The following options are available:
- `-h` Write a help message for invoking ttcp and then exit.
- `-L` Copy the ToolTalk objects of the files, but do not invoke cp(1) to copy the actual files.
- `-p` Preserve. Invoke cp(1) with the `-p` option, which duplicates not only the contents of the original files or directories, but also the modification time and permission modes. The modification times of ToolTalk objects are preserved only if the invoking process has appropriate privileges. (Super-user permissions are required.)
- `-r` Recursively copy the ToolTalk objects of any directories named, along with their files (including any subdirectories and their files), and pass the `-r` option to cp(1).
- `-v` Write the version number of ttcp and then exit.

The `-f`, `-i` or `-R` options to cp(1) are not supported.

OPERANDS
The following operands are supported:
- `filename`
  A pathname of a file to be copied.
- `filename1`
  A pathname of an existing or nonexisting file, used for the output when a single file is copied.
- `directory`
  A pathname of a directory to contain the copied files.
- `directory1`
  A pathname of a file hierarchy to be copied with `-r`.

modified 11 May 1994
**STDIN**
Not used.

**INPUT FILES**
The input files specified as operands can be of any file type.

**ENVIRONMENT VARIABLES**
The following environment variables affect the execution of `ttcp`:

- **LANG**: Provide a default value for the internationalization variables that are unset or null. If `LANG` is unset or null, the corresponding value from the implementation-specific default locale will be used. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.

- **LC_ALL**: If set to a non-empty string value, override the values of all the other internationalization variables.

- **LC_MESSAGES**: Determine the locale that is used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

- **NLSPATH**: Determine the location of message catalogues for the processing of `LC_MESSAGES`.

**RESOURCES**
None.

**ASYNCHRONOUS EVENTS**
The `ttcp` utility takes the standard action for all signals.

**STDOUT**
When the `−h` option is used, `ttcp` writes to standard output a help message in an unspecified format.
When the `−v` option is used, `ttcp` writes to standard output a version number in an unspecified format.

**STDERR**
Used only for diagnostic messages.

**OUTPUT FILES**
The output files can be of any type.

**EXTENDED DESCRIPTION**
None.

**EXIT STATUS**
The following exit values are returned:

- `0`: All files and ToolTalk objects were copied successfully.
- `>0`: An error occurred or the invoked `cp(1)` command exited with a non-zero value.

**CONSEQUENCES OF ERRORS**
Default.

- `/mountpoint/TT_DB`: The directory used as a database for the ToolTalk objects of files in the file system mounted at `/mountpoint`.

modified 11 May 1994
<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td>USAGE</td>
<td>None.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>None.</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>cp(1), ttmv(1), ttar(1), ttsession(1).</td>
</tr>
</tbody>
</table>
NAME | ttdbck – display, check, or repair ToolTalk databases

SYNOPSIS | ttdbck [ selection opts ] [ diagnosis opts ] [ display opts ] [ repair opts ]
[ data-base-directory ]...

DESCRIPTION | ttdbck is the ToolTalk database maintenance tool. It allows direct inspection of ToolTalk spec data, detection of inconsistencies, and repair of problems.

OPTIONS | 

- `data-base-directory` 
  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.
  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.

Selection options | 

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.

- `-f filename` 
  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.

- `-k objidkey` 
  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.

- `-t type` 
  Restricts the set of specs to be inspected or modified to those with otype `type`. The type name can contain shell-style wildcards which must be escaped to prevent the shell from expanding them.

Diagnosis options | 

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.

- `-b` 
  Check for badly formed specs: those which have no file or type or those which have types not defined in the type database.

- `-x` 
  Check for specs which refer to files that no longer exist.

Display options | 

These options determine which data is printed for each selected spec.

- `-i` 
  Display the object id (including the object id key.)

- `-m` 
  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** Display all the properties and values for each selected spec.

- **a** Display all data (equivalent to specifying **-imp**)

**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 Sample_Otype_Name -Z /export/TT_DB
```
In the `/export/TT_DB` directory, finds all specs that refer to objects of type `Sample_Otype_Name` 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 [-G] [-m DTMOUNTPOINT_value] [-n] [-v] [-?]

DESCRIPTION
rpc.ttdbserverd manages ToolTalk objects created by tt_spec_create(3), and handles certain queries related to the netfiles returned by tt_file_netfile(3). One instance of rpc.ttdbserverd (normally started by inetd) runs on each host that has a local filesystem.

rpc.ttdbserverd serves four purposes:
1. Mapping a spec to its associated file and a file to its associated specs.
2. Mapping a spec to its properties.
3. Mapping a file to a list of sessions with clients having patterns registered in the scope of that file.
4. Answering netfile queries; see tt_file_netfile(3) and tt_host_file_netfile(3).

For each filesystem that rpc.ttdbserverd needs to store information about, it creates a directory called TT_DB at the mountpoint of that file system. In that directory it creates the databases it needs to store its tables and indices. If the partition is not writable, then rpc.ttdbserverd can be told, via partition_map(4), to create the databases in another local partition. If rpc.ttdbserverd is not installed on a particular file server, ToolTalk can be told, via hostname_map(4), to manage that file server’s partitions using the rpc.ttdbserverd on a different host.

OPTIONS
- G Perform garbage collection. This cleans up the TT_DB directories and the associated internal database files.

- m DTMOUNTPOINT_value
Sets the DTMOUNTPOINT environment variable for rpc.ttdbserverd. If there is already an environment variable called DTMOUNTPOINT, -m will override it.

- n Turn off permission checking. Normally the protection of the file passed to tt_spec_create(3) 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.

- v Print out the version number.

- ? Prints out the command usage information.

ENVIRONMENT
DTMOUNTPOINT
If set, the value of this environment variable will be used in place of "/net" in pathnames constructed to answer tt_host_file_netfile(3) queries. This environment variable can also be set by using the -m flag for rpc.ttdbserverd.

TT_PARTITION_MAP
If $TT_PARTITION_MAP is set, it is used in place of /etc/tt/partition_map. See partition_map(4).

modified 19 January 1994
FILES

TT_DB/*

spec and session database files are kept in the TT_DB directory under each disk
partition mount point.

`tt/hostname_map`
Host redirection map. See `hostname_map(4)``.

`/etc/tt/partition_map`
Partition redirection map. See `partition_map(4)``.

SEE ALSO

`ttsession(1)` `tt_file_netfile(3)` `tt_host_file_netfile(3)` `tt_spec_create(3)` `hostname_map(4)`
`partition_map(4)`
NAME  ttmv – move or rename files and inform the ToolTalk service

SYNOPSIS  ttmv [-fL] pathname1 pathname2
           ttmv [-fL] pathname ... directory
           ttmv -h | -v

DESCRIPTION  The ttmv utility invokes mv(1) to move files and directories around in the file system and informs ToolTalk about its actions so that the ToolTalk objects associated with those files and directories can also be moved.

The ttmv utility moves the ToolTalk objects before it moves the files and does not check whether the file-moving operation will succeed before performing the object-moving operation.

OPTIONS  The following options are available:

  -f     Force. Do not report any errors, and pass the -f option to mv(1).
  -h     Write a help message for invoking ttmv and then exit.
  -L     Move the ToolTalk objects of the files, but do not invoke mv(1) to move the actual files.
  -v     Write the version number of ttmv and then exit.

The -i option to cp(1) is not supported.

OPERANDS  The following operands are supported:

    pathname1
           A pathname of a file to be moved.
    pathname2
           A pathname of an existing or nonexisting file, used for the output when a single file is moved.
    directory
           A pathname of a directory to contain the moved files.

STDIN  Not used.

INPUT FILES  The input files specified as operands can be of any file type.

ENVIRONMENT VARIABLES  The following environment variables affect the execution of ttmv:

    LANG     Provide a default value for the internationalization variables that are unset or null. If LANG is unset or null, the corresponding value from the implementation-specific default locale will be used. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.

    LC_ALL   If set to a non-empty string value, override the values of all the other

internationalization variables.

*LC_MESSAGES* Determine the locale that is used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

*NLSPATH* Determine the location of message catalogues for the processing of *LC_MESSAGES*.

---

**RESOURCES**

None.

---

**ASYNCHRONOUS EVENTS**

The *ttmv* utility takes the standard action for all signals.

---

**STDOUT**

When the −h option is used, *ttmv* writes to standard output a help message in an unspecified format.

When the −v option is used, *ttmv* writes to standard output a version number in an unspecified format.

---

**STDERR**

Used only for diagnostic messages.

---

**OUTPUT FILES**

The output files can be of any type.

---

**EXTENDED DESCRIPTION**

None.

---

**EXIT STATUS**

The following exit values are returned:

<table>
<thead>
<tr>
<th>Exit Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>All files and ToolTalk objects were moved successfully.</td>
</tr>
<tr>
<td>&gt;0</td>
<td>An error occurred or the invoked <em>mv(1)</em> command exited with a non-zero value.</td>
</tr>
</tbody>
</table>

---

**CONSEQUENCES OF ERRORS**

Default.

---

**FILES**

*mountpoint/TT_DB* The directory used as a database for the ToolTalk objects of files in the file system mounted at *mountpoint*.

---

**APPLICATION USAGE**

None.

---

**EXAMPLES**

None.

---

**SEE ALSO**

*mv(1), ttsession(1).*
NAME  ttrm – remove files or directories and inform the ToolTalk service

SYNOPSIS  ttrm [-frL] pathname ...

            ttrm -h | -v

DESCRIPTION  The ttrm utility invokes rm(1) to remove files and directories and informs ToolTalk about its actions so that the ToolTalk objects associated with the deleted files and directories can also be deleted.

            The ttrm utility removes the ToolTalk objects before it removes the files and does not check whether the file-removing operation will succeed before performing the object-removing operation.

OPTIONS  The following options are available:

            -f  Force. Do not report any errors, and pass the -f option to rm(1).
            -h  Write a help message for invoking ttrm and then exit.
            -L  Remove the ToolTalk objects of the files or directories, but do not invoke rm(1) to remove the actual files or directories.
            -r  Recursively remove the ToolTalk objects of any directories named, along with their files (including any subdirectories and their files), and pass the -r option to rm(1).
            -v  Write the version number of ttrm and then exit.

            The -i or -R options to rm(1) are not supported.

OPERANDS  The following operand is supported:

            pathname

            A pathname of a file to be removed.

STDIN  Not used.

INPUT FILES  The input files specified as operands can be of any file type.

ENVIRONMENT VARIABLES  The following environment variables affect the execution of ttrm:

            LANG  Provide a default value for the internationalization variables that are unset or null. If LANG is unset or null, the corresponding value from the implementation-specific default locale will be used. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.

            LC_ALL  If set to a non-empty string value, override the values of all the other internationalization variables.

            LC_MESSAGES  Determine the locale that is used to affect the format and contents of diagnostic messages written to standard error and informative
messages written to standard output.

**NLSPATH** Determine the location of message catalogues for the processing of *LC_MESSAGES*.

**RESOURCES** None.

**ASYNCHRONOUS EVENTS**
- **STDOUT** The *ttrm* utility takes the standard action for all signals.
  - When the `-h` option is used, *ttrm* writes to standard output a help message in an unspecified format.
  - When the `-v` option is used, *ttrm* writes to standard output a version number in an unspecified format.
- **STDERR** Used only for diagnostic messages.

**OUTPUT FILES** None.

**EXTENDED DESCRIPTION** None.

**EXIT STATUS** The following exit values are returned:
- 0 All files and ToolTalk objects were removed successfully.
- >0 An error occurred or the invoked *rm*(1) command exited with a non-zero value.

**CONSEQUENCES OF ERRORS** Default.

**FILES**
- `mountpoint/TT_DB` The directory used as a database for the ToolTalk objects of files in the file system mounted at `mountpoint`.

**APPLICATION USAGE** None.

**EXAMPLES** None.

**SEE ALSO** *rm*(1), *ttrmdir*(1), *ttsession*(1).

modified 11 May 1994
NAME
ttrmdir – remove empty directories and inform the ToolTalk service

SYNOPSIS

ttrmdir

SYNOPSIS

ttrmdir [−L] directory ...

ttrmdir −h| −v

DESCRIPTION

The ttrmdir utility invokes rmdir(1) to remove empty directories and informs ToolTalk
about its actions so that the ToolTalk objects associated with the deleted directories can
also be deleted.

The ttrmdir utility removes the ToolTalk objects before it removes the directories and
does not check whether a directory is empty or whether the directory-removing opera-
tion will succeed before performing the object-removing operation.

OPTIONS

The following options are available:

−h Write a help message for invoking ttrmdir and then exit.

−L Remove the ToolTalk objects of the directories, but do not invoke rmdir(1) to
remove the actual directories.

−v Write the version number of ttrmdir and then exit.

The −p option to cp(1) is not supported.

OPERANDS

The following operand is supported:

directory

A pathname of an empty directory to be removed.

STDIN

Not used.

INPUT FILES

The input files specified as operands can be of any file type.

ENVIRONMENT VARIABLES

The following environment variables affect the execution of ttrmdir:

LANG Provide a default value for the internationalization variables that are
unset or null. If LANG is unset or null, the corresponding value from
the implementation-specific default locale will be used. If any of the
internationalization variables contains an invalid setting, the utility
behaves as if none of the variables had been defined.

LC_ALL If set to a non-empty string value, override the values of all the other
internationalization variables.

LC_MESSAGES Determine the locale that is used to affect the format and contents of
diagnostic messages written to standard error and informative mes-
sages written to standard output.

NLSPATH Determine the location of message catalogues for the processing of
LC_MESSAGES.
RESOURCES
None.

ASYNCHRONOUS EVENTS
The ttrmdir utility takes the standard action for all signals.

STDOUT
When the −h option is used, ttrmdir writes to standard output a help message in an unspecified format.
When the −v option is used, ttrmdir writes to standard output a version number in an unspecified format.

STDERR
Used only for diagnostic messages.

OUTPUT FILES
None.

EXTENDED DESCRIPTION
None.

EXIT STATUS
The following exit values are returned:

0 All directories and ToolTalk objects were removed successfully.
>0 An error occurred or the invoked rmdir(1) command exited with a non-zero value.

CONSEQUENCES OF ERRORS
Default.

FILES
/mountpoint/TT_DB The directory used as a database for the ToolTalk objects of files in the file system mounted at /mountpoint.

APPLICATION USAGE
The definition of an empty directory is one that contains, at most, directory entries for dot and dot-dot.

EXAMPLES
None.

SEE ALSO
rmdir(1), ttrm(1), ttsession(1).
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 $OPENWINHOME/share/src/tooltalk/demo/ttsample1. Your OPENWINHOME environment variable must be set to where OpenWindows is installed (typically /opt/openwin or /usr/openwin). 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.

SEE ALSO   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 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.

modified 18 Nov 1992
NAME
ttsession – the ToolTalk message server

SYNOPSIS
ttsession [−hNpsStv] [−E] [−X] [−a level] [−d display] [−c [command]]

DESCRIPTION
The ttsession utility 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 that needs to send or receive a message.

OPTIONS
The following options are available:

−a level
Set the server authentication level. The following level string values are supported:

unix The sender and receiver must have the same user ID.
des The underlying RPC calls use AUTH_DES.

−c [command]
Start a process tree session and run the given command. The ttsession utility sets the environment variable TT_SESSION to the name of this session. Any process started with this variable in the environment defaults to being in this session. If command is omitted, ttsession invokes the shell named by the SHELL environment variable. Everything after −c on the command line is used as the command to be executed.

−d display
Specify an X Windows display. The ToolTalk session will consist of those applications displaying on the named display. The default display is identified by the DISPLAY environment variable.

−E Read in the types from the Classing Engine database. If neither −E nor −X is given, −X is assumed.

−h Write a help message to standard error that describes the command syntax of ttsession, and exit.

−N Maximize the number of clients allowed to connect to (in other words, open procids in) this session by attempting to raise the limit of open file descriptors. The precise number of clients is system-dependent; on some systems this option may have no effect.

−p Write the name of a new process tree session to standard output, and then fork a background instance of ttsession to manage this new session.

−s Silent. Do not write any warning messages to standard error.

−S Do not fork a background instance to manage the ttsession session.

−t Turn on trace mode. See ASYNCHRONOUS EVENTS for how to turn tracing

modified 11 May 1994
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.

- \(-v\) Write the version number to standard output and exit.
- \(-X\) Read in the types from the XDR format databases. (Default)

**OPERANDS**
None.

**STDIN**
Not used.

**INPUT FILES**
The XDR format databases listed by the \(-X\) option are serialized ToolTalk data structures of an unspecified format, except that it is the same as the format of `tt_type_comp(1)` output files.

**ENVIRONMENT VARIABLES**
The following environment variables affect the execution of `ttsession`:

- **CEPATH**
  In Classing Engine mode, this variable tells the Classing Engine where to find the databases that contain ToolTalk types. See `ce_db_build(1)`.

- **DISPLAY**
  If `TT_SESSION` is not set and `DISPLAY` is set, then the value of `DISPLAY` will be used by all ToolTalk clients to identify the `ttsession` process serving their X display. If no such process is running, the ToolTalk service will auto-start one.

  If `ttsession` is run with the \(-d\) option and `DISPLAY` is not set, `ttsession` sets `DISPLAY` to be the value of the \(-d\) option for itself and all processes it forks. This helps ToolTalk clients to find the right X display when they are auto-started by `ttsession`.

- **LANG**
  Provide a default value for the internationalization variables that are unset or null. If `LANG` is unset or null, the corresponding value from the implementation-specific default locale will be used. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.

- **LC_ALL**
  If set to a non-empty string value, override the values of all the other internationalization variables.

- **LC_MESSAGES**
  Determine the locale that is used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

- **NLSPATH**
  Determine the location of message catalogues for the processing of `LC_MESSAGES`.

- **TT_ARG_TRACE_WIDTH**
  Specify the number of bytes of argument and context values to write
when in trace mode. The default is to print the first 40 bytes.

**TTPATH**  
In XDR mode, a colon-separated list of directories that tells ToolTalk where to find the ToolTalk types databases. See [tt_type_comp(1)](tt_type_comp(1)).

**TTSESSION_CMD**  
Specify the shell command to be used by all ToolTalk clients for auto-starting `ttsession`.

The `ttsession` utility creates the following variable when it invokes another process:

**TT_FILE**  
When `ttsession` invokes a tool to receive a message, it copies the file attribute (if any) of the message into this variable, formatted in the same manner as returned by the `tt_message_file` function.

**TT_SESSION**  
The `ttsession` utility uses this variable to communicate its session ID to the tools that it starts. The format of the variable is implementation specific. If this variable is set, the ToolTalk client library uses its value as the default session ID.

**TT_TOKEN**  
Inform the ToolTalk client library that it has been invoked by `ttsession`, so that the client can confirm to `ttsession` that it started successfully. The format of the variable is implementation specific.

A tool started by `ttsession` must ensure that the `TT_SESSION` and `TT_TOKEN` are present in the environment of any processes it invokes.

**RESOURCES**  
None.

**ASYNCHRONOUS EVENTS**  
The `ttsession` utility reacts to two signals. If it receives the `SIGUSR1` signal, it toggles trace mode on or off (see the `-t` option). If it receives the `SIGUSR2` signal, it rereads the types file. The `ttsession` utility takes the standard action for all other signals.

**STDOUT**  
When the `-v` option is used, `ttsession` writes the version number in an unspecified format. When `-p` is used, `ttsession` writes the name of a new process tree session.

**STDERR**  
Used only for diagnostic messages and the help message written by the `-h` option.

**OUTPUT FILES**  
None.

**EXTENDED DESCRIPTION**  
None.

**EXIT STATUS**  
When the `-c` child process exits, `ttsession` exits with the status of the exited child. Otherwise, the following exit values are returned:

- **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. The `ttsession` utility was given invalid command line options, was interrupted by `SIGINT`, or encountered some internal error.
- **2** Collision. Another `ttsession` was found to be serving the session already.

modified 11 May 1994
The `ttsession` utility takes the standard action for all signals.

Since everything after `-c` on the command line is used as the command to be executed, `-c` should be the last option.

Tracing is helpful for seeing how messages are dispatched and delivered, but the output may be voluminous.

None.

`tt_type_comp(1), tt_message_file(3)`. 
NAME

tttar − process files and ToolTalk objects in an archive

SYNOPSIS

```
tttar c| t| x [EfhpSv] [tar®le ] pathname ...
tttar c| t| xfL [EhpRSv] tttarfile [−rename oldname newname] ... ] pathname ...
tttar −h| −help
```  

DESCRIPTION

The tttar utility 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 ToolTalk objects onto (or from) a single archive, called a tttarfile.

- With the L function modifier, tttar does not invoke tar to archive actual files, but instead archives (or extracts) only ToolTalk objects onto (or from) a single archive, called a tttarfile. Since without the L function modifier tttar acts like an ToolTalk-aware tar(1), the description below is phrased as if the L function modifier is in effect. That is, the text refers to tttarfiles instead of tarfiles, and it describes archiving and de-archiving only “the ToolTalk objects of the named files” rather than archiving and de-archiving both “the named files and their ToolTalk objects.”

The actions of tttar are controlled by the first argument, the key, a string of characters containing exactly one function letter from the set ctx, and one or more of the optional function modifiers listed under OPERANDS. Other arguments to tttar are file or directory names that specify which files to archive or extract ToolTalk objects for. By default, the appearance of a directory name refers recursively to the files and subdirectories of that directory.

A file does not have to exist for a ToolTalk 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.

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 tttarfile can be placed there temporarily.

OPTIONS

The following options are available:

- `-h`
- `-help` Write a help message for invoking tttar and then exit.

- `-rename oldname newname`
  Interpret the next two arguments as an oldname and a newname, respectively, and rename any entry archived as oldname to newname. If oldname is a directory, then tttar recursively renames the entries as well. If more than one `-rename` option applies to an entry (because of one or more parent directories being renamed), the most specific `-rename` option applies.

- `-v`
  Write the version number of tttar and then exit.
OPERANDS

The following operands are supported:

key  The key operand consists of a function letter followed immediately by zero or more modifying letters.

The function letter is one of the following:

- **c**  Create a new archive and write the ToolTalk objects of the named files onto it.
- **t**  Write to standard output the names of all the files in the archive.
- **x**  Extract the ToolTalk objects of the named files from the archive. If a named file matches a directory with contents in the archive, this directory is (recursively) extracted. The owner and modification time of the ToolTalk objects are restored (if possible). If no filename arguments are given, the ToolTalk objects of all files named in the archive are extracted.

The following characters can be appended to the function letter. Appending the same character more than once produces undefined results.

- **f**  Use the next argument as the name of the ttar file. If ttarfile is given as `−`, ttar writes to the standard output or reads from the standard input, whichever is appropriate.
- **h**  Follow symbolic links as if they were normal files or directories. Normally, ttar does not follow symbolic links.
- **p**  Preserve. Restore the named files to their original modes, ignoring the present umask value (see umask(2)). The ttar utility also extracts setUID and sticky information for the super-user. This option is only useful with the x function letter, and has no meaning if the L function letter is given.
- **L**  Do not invoke tar(1). This modifier must be used with the f function modifier, since reading and writing an ttar archive directly to or from magnetic tape is unimplemented.
- **R**  Do not recurse into directories. This modifier is valid only with the L function modifier.
- **v**  Verbose. Write to standard error the name of each file processed, preceded by a string indicating the operation being performed, as follows:

<table>
<thead>
<tr>
<th>Key Letter</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>&quot;a&quot;</td>
</tr>
<tr>
<td>x</td>
<td>&quot;x&quot;</td>
</tr>
</tbody>
</table>

The file name may be followed by additional information, such as the size of the file in the archive or file system, in an unspecified format. When used with the t function letter, v writes to standard
output more information about the archive entries than just the name.

The following functions and modifiers are not supported:

- The r and u function letters of tar(1), for incrementally updating an archive.
- 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.
- The w function modifier and the −C option of tar(1), for pausing or changing directories between the files listed on the command line.
- Writing and reading ttar®les (that is, archives produced with the L function modifier) directly to and from magnetic tape.

pathname
A pathname of a regular file or directory to be archived (when the c function letter is used), extracted (x) or listed (t). When pathname is the pathname of a directory, the action applies to all of the files and (recursively) subdirectories of that directory. When the f letter is used in the key operand, the initial pathname operand is interpreted as an archive name, as described previously.

tarfile
A pathname of a regular file to be read or written as an archive of files.

ttarfile
A pathname of a regular file to be read or written as an archive of ToolTalk objects.

STDIN
When the f modifier is used with the t or x function letter and the pathname is −, the standard input is an archive file formatted as described in EXTENDED DESCRIPTION. Otherwise, the standard input is not used.

INPUT FILES
The files identified by the pathname operands are regular files or directories. The file identified by the tarfile operand is a regular file formatted as described in tar(1). The file identified by the ttarfile operand is a regular file formatted as described in EXTENDED DESCRIPTION.

ENVIRONMENT VARIABLES
The following environment variables affect the execution of ttar:

LANG
Provide a default value for the internationalization variables that are unset or null. If LANG is unset or null, the corresponding value from the implementation-specific default locale will be used. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.

LC_ALL
If set to a non-empty string value, override the values of all the other internationalization variables.

LC_MESSAGES
Determine the locale that is used to affect the format and contents of
diagnostic messages written to standard error and informative messages written to standard output.

**NLSPATH**
Determine the location of message catalogues for the processing of `LC_MESSAGES`.

**TZ**
Determine the timezone used with date and time strings.

**RESOURCES**
None.

**ASYNCHRONOUS EVENTS**
The `tttar` utility takes the standard action for all signals.

**STDOUT**
When the `-h` option is used, `tttar` writes to standard output a help message in an unspecified format.
When the `-v` option is used, `tttar` writes to standard output a version number in an unspecified format.
When the `f` modifier is used with the `c` function letter and the pathname is `−`, the standard output is an archive file formatted as described in EXTENDED DESCRIPTION. Otherwise, the standard output is not used.

**STDERR**
The standard error is used for diagnostic messages and the file name output described under the `v` modifier (when the `t` function letter is not used).

**OUTPUT FILES**
Output files are created, as specified by the archive, when the `x` function letter is used.

**EXTENDED DESCRIPTION**
The archive file produced and read by `tttar` is formatted as described in `tar(1)`, with the addition of one extra file named `tttarfile`. (If one of the user files being archived is also named `tttarfile`, the results are unspecified.) The `tttarfile` contains all the ToolTalk spec information for the ToolTalk objects in the other files in the archive. The contents of `tttarfile` are written according to the referenced XDR specification (RFC 1014). The only XDR data types used are:

- **int**
  A four-octet signed integer, most significant octet first

- **string**
  A four-octet unsigned integer length, most significant octet first, followed by the characters of the string, followed by sufficient (0 to 3) residual zero octets to make the total number of octets a multiple of four.

The `tttarfile` starts with two integers. The first is always 1, to mark this as the header record. The second is always 1, indicating this is version 1 of the `tttarfile` format. Any future revisions of the `tttarfile` format should increment the version number so older programs processing the `tttarfile` can diagnose the incompatibility.

The end of the `tttarfile` is a integer 3, marking the end-of-file record.

In between, there is one logical record for each spec. Each logical record starts with an integer 2, marking it as a spec record. Other integer values are reserved for assignment to future data types.
After the record identifier, the spec record contains, in sequence:

1. A string giving the ToolTalk object identifier (objid) of the object represented by the spec
2. A string giving the name of the file (as found in the archive table of contents) that contains the contents of the ToolTalk object represented by the spec
3. A string giving the ToolTalk object type identifier (otid) of the ToolTalk object represented by the spec
4. An integer giving the number of properties for this object

The properties of the object immediately follow the number of properties. Each property consists of:

1. A string giving the name of the property
2. An integer, which is always zero (for historical compatibility)
3. An integer giving the number of values for this property
4. A string for each value

After the values, the next property is found, until all properties for the object have been accounted for; then the next spec is found, until all specs for objects associated with files in the archive are accounted for.

**EXIT STATUS**

The following exit values are returned:

- 0 All files and ToolTalk objects were moved successfully.
- >0 An error occurred or the invoked `tar(1)` command exited with a non-zero value.

**CONSEQUENCES OF ERRORS**

Default.

**FILES**

`/mountpoint/TT_DB` The directory used as a database for the ToolTalk objects of files in the file system mounted at `/mountpoint`.

**APPLICATION USAGE**

None.

**EXAMPLES**

None.

**SEE ALSO**

`tar(1), ttcp(1), ttsession(1)`. 

modified 11 May 1994
NAME
ttrace – trace ToolTalk calls and messages

SYNOPSIS
ttrace [-0FCa] [-o outfile | [-S session | command]
ttrace [-e script | -f scriptfile] [-S session | command]

DESCRIPTION
ttrace traces message traffic through the server for the indicated ToolTalk
session, or runs command with ToolTalk client tracing turned on. If neither session nor command is given,
the default session is traced. By default, tracing terminates when tttrace exits.

Tracing of ToolTalk functions looks like this:
[pid] function_name(params) = return_value (Tt_status)

With the -a option, message attributes are printed after a one-line summary of the mes-
sage:
Tt_state Tt_paradigm Tt_class (Tt_disposition in Tt_scope): status == Tt_status

State changes are indicated by:
old_state => new_state.

Deliveries are indicated by:
Tt_message => procid <recipient_procid>

When dispatching is being traced, the reason for each dispatch is one of:

tt_message_send()
tt_message_reject()
tt_message_fail()
tt_message_reply()
tt_session_join()
tt_file_join()
tt_message_reply()

A client called the indicated function.

tt_message_send_on_exit()

ttsession is dispatching on_exit messages for a client that disconnected before
calling tt_close().

tt_message_accept()

ttsession is dispatching messages that had been blocked while a ptype was
being started. The started client has now called either tt_message_accept() or

TT_ERR_PTYPE_START
A ptype instance was started to receive the message, but the start command
exited before it connected to ttsession.

TT_ERR_PROCID

ttsession lost its connection to the client that was working on this request.

Another session wants this session to find recipients for the message.

ttsession <- ttsession

modified 19 January 1994
Another session wants to update (e.g., fail) a message originating in this session.

When dispatching is being traced, matching is indicated by one of

\[
\begin{align*}
&Tt_{\text{message}} \& Tt_{\text{pattern}} \\
&Tt_{\text{message}} \& ptype \ \text{ptid} \\
&Tt_{\text{message}} \& otype \ \text{otid}
\end{align*}
\]

The pattern or signature is printed, followed by

\[\text{match_score}; \quad \text{[\text{mismatch_reason}]}/\ast\]

**OPTIONS**

- **-0**  
  Turn off message tracing in `session`, or run `command` without message tracing (i.e., with only call tracing).

- **-F**  
  Follow all children forked by `command` or subsequently started in `session` by `ttsession`. Normally, only the indicated `command` or `ttsession` instance is traced. When `-F` is specified, the process id is included with each line of trace output to indicate which process generated it.

- **-C**  
  Do not trace client calls into the ToolTalk API. Default is to trace them.

- **-a**  
  Print all attributes, arguments, and context slots of traced messages. The default is to use only a single line when printing a message on the trace output.

- **-e script**  
  Take `script` as a `tttrace` setting. See `tttracefile(4)`.

- **-f scriptfile**  
  File to read `tttrace` settings from. See `tttracefile(4)`. `-f` causes `tttrace` to read standard input until EOF, which may prevent `command` from using standard input.

- **-o outfile**  
  File to be used for the trace output.

  For session tracing, output goes to standard output of `tttrace`.

  For client tracing, output goes by default to standard error of `tttrace`. For client tracing, `-o` causes trace output to go to standard output of `tttrace`.

  If the server for `session` is running on a remote host and either
  
  - `outfile` is not mounted on that host, or
  - the `-o` option is omitted,
  
  then `tttrace` will fail.

- **-S session**  
  Session to trace. Defaults to the default session -- the session that `tt_open(0)` would contact.

- **command**  
  The ToolTalk client command to invoke and trace.

**EXAMPLES**

Here we trace a client that registers a pattern and sends a notice that matches it:

```
% tttrace -a myclientprogram
```
tt_open() = 0x51708="7.jOHHM X 129.144.153.55 0" (TT_OK)
tt_fd() = 11 (TT_OK)
tt_pattern_create() = 0x50318 (TT_OK)
tt_pattern_category_set(0x50318, TT_OBSERVE) = 0 (TT_OK)
tt_pattern_scope_add(0x50318, TT_SESSION) = 0 (TT_OK)
tt_pattern_op_add(0x50318, 0x2f308=="Hello World") = 0 (TT_OK)
tt_default_session() = 0x519e0="X 129.144.153.55 0" (TT_OK)
tt_pattern_session_add(0x50318, 0x519e0="X 129.144.153.55 0") = 0 (TT_OK)
tt_pattern_register(0x50318) = 0 (TT_OK)
tt_message_create() = 0x51af0 (TT_OK)
tt_message_class_set(0x51af0, TT_NOTICE) = 0 (TT_OK)
tt_message_address_set(0x51af0, TT_PROCEDURE) = 0 (TT_OK)
tt_message_scope_set(0x51af0, TT_SESSION) = 0 (TT_OK)
tt_message_op_set(0x51af0, 0x2f308=="Hello World") = 0 (TT_OK)
tt_message_send(0x51af0) ...
    TT_CREATED => TT_SENT:
    TT_SENT TT_PROCEDURE TT_NOTICE (TT_DISCARD in TT_SESSION): 0 ==
    id: 0 7.jOHHM X 129.144.153.55 0
    op: Hello World
    session: X 129.144.153.55 0
    sender: 7.jOHHM X 129.144.153.55 0
    = 0 (TT_OK)

tt_message_receive() ...
    Tt_message => procid <7.jOHHM X 129.144.153.55 0>
    TT_SENT TT_PROCEDURE TT_NOTICE (TT_DISCARD in TT_SESSION): 0 == TT_OK
    id: 0 7.jOHHM X 129.144.153.55 0
    op: Hello World
    session: X 129.144.153.55 0
    sender: 7.jOHHM X 129.144.153.55 0
    pattern: 0:7.jOHHM X 129.144.153.55 0
    = 0x51af0 (TT_OK)

 ttsession’s view of this traffic can be seen as follows. Note that the first message traced
will almost always be ttsession’s reply to the request sent it by tttrace.
% tttrace -a

tt_message_reply:
    TT_SENT => TT_HANDLED:
    TT_HANDLED TT_PROCEDURE TT_REQUEST (TT_DISCARD in TT_SESSION): 0 ==
    id: 0 2.jOHHM X 129.144.153.55 0
    op: Session_Trace
    args:
    TT_IN string: ">
    session: X 129.144.153.55 0
    sender: 2.jOHHM X 129.144.153.55 0
    pattern: 0:2.jOHHM X 129.144.153.55 0

modified 19 January 1994
To trace message flow in a specific, non-default session,

```bash
% tttrace -S '01 15303 1342177284 4 1 0 13691 129.144.153.55 2'
```

**ENVIRONMENT**

`tttrace` is implemented purely as a ToolTalk client, using the message interface to `ttsession` and the following environmental hook into `libtt`.

**TT_TRACE_SCRIPT**

If set, tells `libtt` to turn on client-side tracing as specified in the trace script. If the first character of the value is `.` or `/`, the value is taken to be the pathname of file containing the trace script to use. Otherwise, the value is taken to be an inline trace script.

**FILES**

`$TMPDIR/tttrace.nnn`

A named pipe (see `mkfifo(3C)`) in `$TMPDIR` (see `tempnam(3S)`) from which trace output for `session` is read when the `-o` option is omitted.

**WARNINGS**

Since (with the `-F` option) tracing can follow clients to remote hosts if the environment is properly propagated, it is possible for different processes in the same trace output to be labeled with the same process id.

---

modified 19 January 1994
SEE ALSO

ttsession(1), tttracefile(4), the Session_Trace() ToolTalk request

DIAGNOSTICS

If command is run, then tttrace will exit with the exit status of command. Otherwise, exit codes are as follows:

0  Normal termination. Any session tracing turned on by this invocation of tttrace has now been turned off.
1  Usage. tttrace was given invalid command line options.
2  Failure. tttrace encountered an error while trying to do its job. An error message has been emitted on standard error.
3  Runaway session tracing. tttrace could not terminate tracing in session before exiting.
4  Remote session. ttsession is remote, and outfile (if given) is not visible there. Choose a visible file, or run tttrace on that remote host.
5  Old session. The ttsession for session does not support the Session_Trace() request. Run kill -USR1 on it to turn on old-style tracing.

NOTES

For security purposes, client-side tracing is disabled inside a client when its effective uid or gid is different from its real uid or gid and the real uid is not the super-user.
NAME   tt_type_comp - compile ToolTalk otypes and ptypes

SYNOPSIS  tt_type_comp [-mMs] [-d db] source_file
          tt_type_comp -r [-s] [-d db] type ...
          tt_type_comp -p | -O | -P [-sE] [-d db]
          tt_type_comp -p | -O | -P [-s] source_file
          tt_type_comp -x [-s] [-o compiled_file] source_file
          tt_type_comp [-hv]

DESCRIPTION  The tt_type_comp utility processes otypes and ptypes. 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. The tt_type_comp utility
              preprocesses the source types with cpp(1), and can optionally write out the compiled
types instead of merging them into the standard databases. The tt_type_comp utility can
also remove types from the standard databases or write out the contents of these databases.

The tt_type_comp utility 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 Class-
ing 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).

OPTIONS  The following options are available:
          -d db  Specify the database to work on, which must be one of user, system or net-
                  work. For Classing Engine mode these are defined as:
                  user       $HOME/.cetables/cetables
                  system     /etc/cetables/cetables
                  network    $OPENWINHOME/lib/cetables/cetables
          For the XDR format these are defined respectively as the first, second, and last
          elements of $TTPATH.
          These 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 -p, -O and -P
options, the default is all three databases.

          -E  Use Classing Engine mode, instead of the default XDR mode.
          -h  Write a help message for invoking tt_type_comp and then exit.
          -m  Merge types into the specified database, updating any existing type with the
               new definition given. This is the default action. This action is not supported
               for Classing Engine mode.
−M  Merge types into the specified database (see −m), but only if they do not already exist in that database. This action is not supported for Classing Engine mode.

−O  Write the names of all otypes read.

−p  Write the ToolTalk types read in a source format suitable for recompilation with tt_type_comp.

−P  Write the names of all ptypes read.

−o  compiled_file
   Write the compiled types into the specified file, or to standard output if compiled_file is −.

−r  Remove the given ptypes or otypes from the specified database, as indicated by the type operands.

−s  Silent mode. Write nothing to standard output.

−v  Write the version number of tt_type_comp and then exit.

−x  Compile source types into a compiled types file, instead of merging them into the standard types databases.

OPERANDS  The following operands are supported:

source_file
   A pathname of a text file containing ToolTalk source code. If source_file is −, standard input is used.

type
   A name of a type to be removed by the −r option.

STDIN  The standard input is used only if a source_file operand is −.

INPUT FILES  The input file named by source_file is a text file containing ToolTalk source code.

ENVIRONMENT VARIABLES  The following environment variables affect the execution of tt_type_comp:

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).

LANG  Provide a default value for the internationalization variables that are unset or null. If LANG is unset or null, the corresponding value from the implementation-specific default locale will be used. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.

LC_ALL  If set to a non-empty string value, override the values of all the other internationalization variables.

LC_MESSAGES  Determine the locale that is used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.
OpenWindows Desktop 3.5		User Commands	tt_type_comp (1)

NLSPATH	Determine the location of message catalogues for the processing of LC_MESSAGES.

TTPATH	In XDR mode, a colon-separated list of directories that tells the ToolTalk service where to find the ToolTalk types databases. If TTPATH has no value or is not set, it is considered to be:

$HOME/.tt:
/etc/tt:
/usr/dt/appconfig/tttypes:
$OPENWINHOME/etc/tt

RESOURCES	None.

ASYNCHRONOUS EVENTS

STDOUT	The tt_type_comp utility takes the standard action for all signals.

When the −h option is used, tt_type_comp writes to standard output a help message in an unspecified format.

When the −o option is used, tt_type_comp writes to standard output a listing of all otypes read.

When the −p option is used, tt_type_comp writes to standard output a listing of all the ToolTalk types read, in a source format suitable for recompilation with tt_type_comp.

When the −P option is used, tt_type_comp writes to standard output a listing of all ptypes read.

When the −v option is used, tt_type_comp writes to standard output a version number in an unspecified format.

STDERR	Used only for diagnostic messages.

OUTPUT FILES	When the −x or −d user option is used, tt_type_comp writes the compiled types in an unspecified format into a user-specified file. Otherwise, it writes the compiled types into the databases described under −d.

EXTENDED DESCRIPTION

EXIT STATUS	The following exit values are returned:

0	Successful completion.
1	Usage; tt_type_comp was given invalid command line options.
2	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.

CONSEQUENCES OF ERRORS	Default.

modified 11 May 1994
FILES

$HOME/tt/types.xdr
    User's ToolTalk types database for XDR mode

/etc/tt/types.xdr
    System ToolTalk types database for XDR mode

/usr/dt/appconfig/tttypes/types.xdr

$OPENWINHOME/etc/tt/types.xdr
    Network ToolTalk types databases 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).

APPLICATION

USAGE
    None.

EXAMPLES
    None.

SEE ALSO
    ttsession(1), ce_db_build(1), ce_db_merge(1), cpp(1).
**NAME**
viewprint – print AnswerBook documents using lp

**SYNOPSIS**
viewprint -P printer | -f file [ -# ncopies ] [ -p pages ] [ -R ] [ -c card-catalog-file ]
document-name

**DESCRIPTION**
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 document-name, and prints these files using lp(1).

OpenWindows users do not typically need to use viewprint. It is automatically invoked by docviewer as needed.

**OPTIONS**
- `-P printer`
  Send document to printer
- `-f filename`
  Send document to filename (instead of printer)
- `-# ncopies`
  Print ncopies copies
- `-p pages`
  Specify page numbers of pages to print. pages can be a single page number, or a page range of the form first-last
- `-R`
  Reverse the order of pages (last page first)
- `-c card-catalog`
  Specify the name of the card catalog file used to locate AnswerBooks. See ab_cardcatalog(4) for more information.

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

**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**
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.

PostScript is a trademark of Adobe Systems Incorporated.
AnswerBook is a trademark of Sun Microsystems, Inc., licensed to SunSoft, Inc.
NAME   vkbd – virtual keyboard/soft function keys display manager

SYNOPSIS  vkbd [-nopopup] [generic-tool-arguments]

DESCRIPTION  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.

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:

OpenWindows.KeyboardCommand.Translate: key

vkbd is usually started from openwin-sys and is not intended to be used directly.

OPTIONS  

-nopopup

This option indicates that Function Keys popup panel is not mapped at startup time.

FILES  $OPENWINHOME/lib/openwin-sys

SEE ALSO  xview(7)

OPENLOOK Graphical User Interface Functional Specification

modified 24 March 1992
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 XView Programming Manual and the XView Reference Manual.

USAGE
Compiling XView Programs
XView programs are compiled with the following command line:

cc sourcefile.c -o outputfile -lxview -lolgx -lx11

Generic XView Functions

**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.

```
Xv_Server
xv_init(attrs)
<attribute-value list> attrs;
```

Note that `attrs` is a NULL terminated attribute-value list.

**xv_create()**
Creates an object.

```
Xv_object
xv_create(owner, package, attrs)
Xv_object owner;
Xv_pkg package;
<attribute-value list> attrs;
```

**xv_destroy()**
Destroys an object.

```
int
xv_destroy(object)
Xv_opaque object;
```

**xv_find()**
Finds an object that meets certain criteria; or if the object doesn’t exist,
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 |

modified 24 December 1991
**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` command. If the font you specify cannot be found, you will see an error message such as: XView warning: Cannot load font set ‘galant-24’ (Font package)  
**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.
### 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:**

```plaintext
Example: cmdtool -geometry 500x600
(will make the base frame 500x600 pixels, with the position set by the window manager)

Example: 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)

Example: 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)

Example: 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:**

```plaintext
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 back-  
ground 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 con-  
struct the resource name used to perform lookups in the X11 Resource  
Manager to look for the values of customizable attributes.

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_basiclocale

**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 con-
ditions. 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 fol-
lowing entry to the X resource database:
Window PassiveGrab Select False
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 behavior 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:

<table>
<thead>
<tr>
<th>Resource:</th>
<th>Resource Name (If the resource can be modified by the OpenWindows Property Sheet, the word Props will be present.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>Possible Values, and/or Format of Values to be Assigned to Resource (Default Value)</td>
</tr>
<tr>
<td>Description:</td>
<td>Description of Resource.</td>
</tr>
</tbody>
</table>

---

**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.icccmcompliant
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.
<table>
<thead>
<tr>
<th>Resource:</th>
<th>Selection.Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>N (3)</td>
</tr>
<tr>
<td>Description:</td>
<td>Selection timeout value. N indicates the number of seconds that a requestor or a selection owner waits for a response.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>OpenWindows.GotoMenu.RecentCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>integer between 0 and 15 (8)</td>
</tr>
<tr>
<td>Description:</td>
<td>Specifies the number of recently visited directories shown in the Go To Menu of a File Chooser.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>string-list (NULL)</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>These resources determine mouseless semantic action and its corresponding key binding. Refer to the XView Reference Manual for a complete listing and explanation of the OpenWindows.KeyboardCommand.∗ resources. Refer to the XView Programming Manual for information on the mouseless model.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>OpenWindows.KeyboardCommands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>SunView1, Basic, or Full</td>
</tr>
<tr>
<td>Description:</td>
<td>Controls the level of mouseless operation. All of the OpenWindows.KeyboardCommand resource mappings may be modified by users, or by specifying one of the three values for OpenWindows.KeyboardCommands. For detailed information see the XView Programming Manual.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>OpenWindows.MenuAccelerators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True or False (True)</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>OpenWindows.MouseChordMenu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (False)</td>
</tr>
<tr>
<td>Description:</td>
<td>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.</td>
</tr>
<tr>
<td>Resource:</td>
<td>OpenWindows.MouseChordTimeout</td>
</tr>
<tr>
<td>Values:</td>
<td>N (100)</td>
</tr>
<tr>
<td>Description:</td>
<td>Mouse chording time-out value. N is in micro-seconds.</td>
</tr>
</tbody>
</table>

| 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!"#$%&'()*+,-./:;<=>?@[\]^\_`{|}Ä') |
| 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 modified 24 December 1991 |
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: | scrollbar.jumpCursor (Props) |
| Values: | True, False (True) |
| Description | When False, the scrollbar will not move the mouse pointer when scrolling. |

| Resource: | scrollbar.repeatDelay |
| 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: | scrollbar.pageInterval |
| 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: | scrollbar.lineInterval |
| 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: | text.maxDocumentSize |
| 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: | text.retained |
| Values: | True, False (False) |
| Description | If True, retain text windows with server backing store. |

| Resource: | text.extrasMenuFilename |
| Values: | filename (/usr/lib/.text_extras_menu) |
| 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.

<table>
<thead>
<tr>
<th>Resource</th>
<th>text.enableScrollbar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description</td>
<td>When False, do not put a scrollbar on textsw objects.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>text.againLimit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>N (1)</td>
</tr>
<tr>
<td>Description</td>
<td>Where N is an integer between 0 and 500. Number of operations the &quot;again history&quot; remembers for a textsw.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>text.autoIndent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (False)</td>
</tr>
<tr>
<td>Description</td>
<td>When True, begin the next line at the same indentation as the previous line as typing in text.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>text.autoScrollBy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>N (1)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>text.confirmOverwrite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description</td>
<td>When False, do not give user confirmation if a save will overwrite an existing file.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>text.displayControlChars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>Text.DeleteReplacesClipboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (False)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Resource:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>text.undoLimit</td>
<td>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 &quot;Undo&quot; key in the text window.</td>
</tr>
<tr>
<td>text.insertMakesCaretVisible</td>
<td>Controls whether insertion causes repositioning to make inserted text visible.</td>
</tr>
<tr>
<td>text.lineBreak</td>
<td>Determines how the textsw treats file lines when they are too big to fit on one display line.</td>
</tr>
<tr>
<td>text.margin.bottom</td>
<td>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.</td>
</tr>
<tr>
<td>mouse.multiclick.space</td>
<td>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.</td>
</tr>
<tr>
<td>text.storeChangesFile</td>
<td>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.</td>
</tr>
<tr>
<td>text.margin.top</td>
<td>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.</td>
</tr>
<tr>
<td>text.margin.left</td>
<td>Where N is an integer between 0 and 2000. Specifies the margin in</td>
</tr>
</tbody>
</table>
pixels that the text should maintain between the left hand border of the window and the first character on each line.

<table>
<thead>
<tr>
<th>Resource:</th>
<th>text.margin.right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>N (0)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>text.tabWidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>N (8)</td>
</tr>
<tr>
<td>Description</td>
<td>Where N is an integer between 0 and 50. Specifies the width in characters of the tab character.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>Text.LineSpacing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>N (0)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>term.boldStyle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>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)</td>
</tr>
<tr>
<td>Description</td>
<td>Specify the text bolding style for a terminal based window.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>term.inverseStyle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>Enable, Disable, Same_as_bold (Enable)</td>
</tr>
<tr>
<td>Description</td>
<td>Specify the text inverting style for a terminal based window.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>term.underlineStyle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>Enable, Disable, Same_as_bold (Enable)</td>
</tr>
<tr>
<td>Description</td>
<td>Specify the text underlining style for a terminal based window.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource:</th>
<th>term.useAlternateTtyswrc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>True, False (True)</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

modified 24 December 1991
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.
$HELP PATH 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

modified 24 December 1991
SOPENWINHOME/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 seconds 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
OpenWindows documentation

modified 17 March 1992
Index

Special Characters
24to8(1) — converts Sun raster®les to 8-bit format, 1

A
ab_admin(1) — administer AnswerBook card catalog database(s), 1
ab_cardcatalog(4) — file of available AnswerBooks, 1
ab_library(4) — library of AnswerBooks searched by navigator — ab_library(4), 1
align_equals — textedit selection filter, 1
answerbook(1) — on-line documentation viewing system, 1
AnswerBook
administer card catalog database(s) — ab_admin(1), 1
browse and search AnswerBook documentation — navigator(1), 1
documentation viewer — docviewer(1), 1
file of available AnswerBooks — ab_cardcatalog(4), 1
general information — answerbook(1)
library of AnswerBooks searched by by navigator — ab_library(4), 1
print AnswerBook documents using lp — viewprint(1), 1
audiocontrol(1) — audio control tool, 1
audiotool(1) — audio recording & playback tool, 1

B
binder(1) — modify deskset bindings database, 1

C
calctool(1) — desktop calculator tool
Calendar Manager
browse calendar appointments —
cm_lookup(1), 1
deamon used with cm — rpc.cmsd(1), 1
delete calendar appointments — cm_delete(1), 1
general information — cm(1)
insert calendar appointments — cm_insert(1), 1
capitalize — textedit selection filter, 1
ce_db_build(1) — build CE database from ASCII files, 1
ce_db_merge(1) — merge new data into CE database, 1
clock(1) — desktop clock program, 1
cm(1) — desktop calendar tool, 1
cm_delete(1) — delete calendar appointments from command line, 1
cm_insert(1) — inserts calendar appointments from command line, 1
cm_lookup(1) — browse calendar appointments from command line, 1
cmdtool(1) — enhanced terminal window program,
color
converting Sun raster files to 8-bit format —
24to8(1), 1
icon color manipulation tool — colorchooser(1), 1
utility to reduce colormap flashing —
ds_server_init(1), 1
colorchooser(1) — icon color manipulation tool, 1
convert_to_Xdefaults(1) — SunView settings converter, 1
convert_to_xview(1) — SunView to Xview converter, 1

demonstration programs
client/server demo programs — edit_demo(6), 1
fishtank animation demo — realxfishdb(6), 1
OLIT demo program — olitsampler(6), 1
simple ToolTalk demo — tsample(6), 1
docviewer(1) — viewer for AnswerBook documentation, 1
ds_server_init(1) — utility to reduce colormap flashing, 1
dsdm(1) — drop site database manager, 1

edit_demo(6) — ToolTalk client/server demo programs, 1

filemgr(1) — file management tool, 1
fixframe — converts FrameMaker *.ps files to
PostScript, 1
fmgc(1) — clean up files left by filemgr — fmgc(1), 1
format_floppy(1) — diskette formatting program, 1

helpopen(1) — program to communicate with helpviewer, 1
helpviewer(1) — OpenWindows viewer for on-line help, 1

iconedit(1) — DeskSet utility for editing icons, 1
images
capturing screens — snapshot(1), 1
OpenWindows viewing program — imagetool(1), 1
rasterfile conversion utility — rash(1), 1
imagetool(1) — Image viewing application, 1
insert_brackets — textedit selection filter, 1

keyboard
virtual keys & function displayer — vkbd(1), 1

locale_env(1) — sets language, 1

mail
OLIT mailbox monitor — olbiff(6), 1
OPEN LOOK interface for MH — olmh(6)
OpenWindows mail interface — mailtool(1), 1
removing attachments — mailprint(1), 1
mailprint(1) — mail message attachment filter, 1
mailtool(1) — OpenWindows mail interface, 1
mp(1) — PostScript printer program, 1

navigator(1) — browse and search AnswerBook documentation, 1

olbiff(6) — OLIT mailbox monitor utility, 1
olitsampler(6) — OLIT widget demo, 1
olittable(6) — periodic table demo, 1
olmh(6) — OPEN LOOK interface for MH system
olwm(1) — OPEN LOOK window manager
olwmslave(1) — olwm utility program, 1
openwin(1) — OpenWindows startup script, 1
OpenWindows
audio control tool — audiocontrol(1), 1

Index–2
OpenWindows, continued

- calculator tool — calc_tool(1), 1
- client listing utility — ow_places(1), 1
- clock program — clock(1), 1
- file management tool — filemgr(1), 1
- mail interface — mailtool(1), 1
- on-line help viewer — helpviewer(1), 1
- OPEN LOOK window manager — olwm(1)
- printing tool — printtool(1), 1
- property setting utility — props(1), 1
- screen capturing program — snapshot(1), 1
- shell window program — shelltool(1), 1
- startup script — openwin(1), 1
- tape drive control tool — tapetool(1), 1
- ow_places(1) — X client listing utility, 1

---

P

- pageview(1) — PostScript language previewer, 1
- perfmeter(1) — system performance display program, 1
- PostScript
  - convert FrameMaker *.ps files — fix_frame(1), 1
  - converting rasterfiles — rash(1), 1
  - printing program — mp(1), 1
- printtool(1) — OpenWindows printing tool, 1
- props(1) — OpenWindows property setting utility, 1

---

R

- rash(1) — rasterfile to PostScript translator, 1
- realxfishdb(6) — animated fishtank display, 1
- remove_brackets — textedit selection filter, 1
- rpc.cmsd(1) — calendar manager service deamon, 1

---

S

- shelltool(1) — OpenWindows shell program, 1
- shift_lines — textedit selection filter, 1
- snapshot(1) — screen image capture tool, 1
- spider(6) — double-deck solitaire card game, 1

---

T

- tapetool(1) — tape drive control tool, 1
- text
  - command window — cmdtool(1), 1
  - filters for text processing — textedit_filters(1), 1
- xview(7) — xview toolkit information, 1
- ToolTalk
  - client/server demo programs — edit_demo(6), 1
  - simple ToolTalk demo — ttsample(6), 1
- toolwait(1) — X client startup utility, 1
- ttsample(6) — simple ToolTalk demo, 1

---

V

- viewprint(1) — print AnswerBook documents using lp, 1
- viewprint command, 1
- vkbd(1) — virtual keys & functions display, 1

---

W

- windows
  - changing fonts or colors — props(1), 1
  - OPEN LOOK window manager — olwm(1)
  - shell window program — shelltool(1), 1
  - terminal window program — cmdtool(1), 1

---

X

- xv_get_sel(1) — copy selection to standard output, 1
- xview(7) — xview toolkit information, 1

---

Index-3