NAME

olmh – OpenLook interface to the MH message handling system

SYNOPSIS

olmh [**-path** *mailpath*] [**-initial** *foldername*] [**-flag**] [*-toolkitoption...*]

DESCRIPTION

The **olmh** 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 **-path** followed by the pathname of the alternate mail directory. The default mail path is the value of the Path component in $HOME/.mh_profile, or $HOME/Mail if the *MH* Path is not given. To specify an alternate folder which may receive new mail and is initially opened by **olmh**, use the **-initial** flag. The default initial folder is 'inbox'. The option **-flag** will cause **olmh** to attempt to change the appearance of its icon when new mail has arrived. These three options have corresponding application-specific resources, named **MailPath**, **InitialFolder**, and **MailWaitingFlag**, which can be used in a resource file. The standard toolkit command line options are given in **X11**(7).

Please don't be misled by the size of this document. It introduces many aspects of the OLIT Widget Set, and provides extensive mechanism for customization of the user interface. **olmh** really is easy to use.

INSTALLATION

The current version of **olmh** 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 **olmh** 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 **olmh** to incorporate new mail.

For more information, refer to the **mh**(1) documentation.

BASIC SCREEN LAYOUT

olmh starts out with a single window, divided into four main areas:

– Five buttons with pull-down command menus.
– A collection of buttons, one for each top level folder. New users of mh will have two folders, ''drafts'' and ''inbox''.
– A listing, or Table of Contents, of the messages in the open folder. Initially, this will show the messages in ''inbox''.

– A view of one of your messages. Initially this is blank.

OLMH AND THE OLIT WIDGET SET

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.

HIGHLIGHTED MESSAGES, SELECTED MESSAGES, CURRENT MESSAGE

It is possible to highlight a message in the area of the Table of Contents. To highlight a message, click on it with pointer button 1.

The selected message is the same as the highlighted message, if any. If no message is highlighted, then the selected message is considered the same as the current message.

The current message is indicated by a '+' next to the message number. It usually corresponds to the message currently being viewed. When a message is viewed, the title bar above the view will identify the message.

TABLE OF CONTENTS COMMANDS

The *Table of Contents* command menu contains commands which operate on the open, or viewed folder.

Incorporate New Mail
Add any new mail received to your inbox folder, and set the current message to be the first new message. (This command is selectable only if ''inbox'' is the folder being viewed.)

Commit Changes Execute all deletions, moves, and copies that have been marked in this folder.

Pack Folder Renumber the messages in this folder so they start with 1 and increment by 1.

Sort Folder Sort the messages in this folder in chronological order. As a side effect, this also packs the folder.

Rescan Folder Rebuild the list of messages. This can be used whenever you suspect that **olmh**'s idea of what messages you have is wrong. (In particular, this is necessary if you change things using straight *MH* commands without using **olmh**.)

Read in Reverse Read in Reverse will switch the meaning of the next and previous messages, and will increment in the opposite direction. This is useful if you want to read your messages in the order of most recent first. The option acts as a toggle; select it from the menu a second time to undo the effect.

MESSAGE COMMANDS

The *Message* command menu contains commands which operate on the selected message(s), or if there are no selected messages, the current message.

Compose Message Composes a new message. A new window will be brought up for composition; a description of it is given in the COMPOSITION WINDOWS section below. This command does not affect the current message.

View Next Message
View the first selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message after the current message.

View Previous View the last selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message before the current message.

Mark Deleted Mark the selected messages for deletion. If no messages are highlighted, then this will mark the current message for deletion and automatically display the next unmarked message.

Mark Move Mark the selected messages to be moved into the current (selected) folder. (If the current folder is the same as the viewed folder, this command will just beep.) If no messages are highlighted, this will mark the current message to be moved and display the next unmarked message.

Mark Copy Mark the selected messages to be copied into the current folder. (If the current folder is the same as the viewed folder, this command will just beep.) If no messages are highlighted, mark the current message to be copied.

Unmark Remove any of the above three marks from the selected messages, or the current message, if none are highlighted.

View in New Window
Create a new window containing only a view of the first selected message, or the current message, if none are highlighted.

Reply Create a composition window in reply to the first selected message, or the current message, if none are highlighted.

Forward Create a composition window whose body is initialized to be the contents of the selected messages, or the current message if none are highlighted.

Use as Composition
Create a composition window whose body is initialized to be the contents of the first selected message, or the current message if none

are selected. Any changes you make in the composition will be saved in a new message in the ''drafts'' folder, and will not change the original message. However, this command was designed to be used within the ''drafts'' folder to compose message drafts, and there is an exception to this rule. If the message to be used as composition was selected from the ''drafts'' folder, the changes will be reflected in the original message (see COMPOSITION WINDOWS).

Print Print the selected messages, or the current message if none are selected. **olmh** normally prints by invoking the **enscript**(1) command, but this can be customized with the application-specific resource **PrintCommand**.

SEQUENCE COMMANDS

The *Sequence* command menu of xmh allows the user to view chosen sequences of messages. Unfortunately this useful feature is not available in this demonstration version of Olmh.

VIEW COMMANDS

Commands in the View menu and in the buttonboxes of view windows (which result from the Message command ''View In New'') correspond in functionality to commands of the same name in the Message menu, but they operate on the viewed message rather than the selected messages or current message.

Close Window When the viewed message is in a separate view window, this command will close the view, after confirming the status of any unsaved edits.

Reply Create a composition window in reply to the viewed message.

Forward Create a composition window whose body is initialized to be the contents of the viewed message.

Use As Composition
Create a composition window whose body is initialized to be the contents of the viewed message. Any changes made in the composition window will be saved in a new message in the ''drafts'' folder, and will not change the original message. An exception: if the viewed message was selected from the ''drafts'' folder, the original message is edited.

Edit Message This command enables the direct editing of the viewed message.

Save Message This command is insensitive until the message has been edited; when activated, edits will be saved to the original message in the view.

Print Print the viewed message. **olmh** prints by invoking the **enscript**(1) command, but this can be customized with the application-specific resource **PrintCommand**.

COMPOSITION WINDOWS

Aside from the normal text editing functions, there are six command buttons associated with composition windows:

Close Window	Close this composition window. If changes have been made since the most recent Save or Send, you will be asked to confirm losing them.
Send	Send this composition.
New Headers	Replace the current composition with an empty message. If changes have been made since the most recent Send or Save, you will be asked to confirm losing them.
Compose Message	Bring up another new composition window.
Save Message	Save this composition in your drafts folder. Then you can safely close the composition. At some future date, you can continue working on the composition by opening the drafts folder, selecting the message, and using the ''Use as Composition'' command.
Insert	Insert a related message into the composition. If the composition window was created with a ''Reply'' command, the related message is the message being replied to, otherwise no related message is defined and this button is insensitive. The message may be filtered before being inserted; see **ReplyInsertFilter** under APPLICATION RESOURCES for more information.

ACCELERATORS

Accelerators are shortcuts. They allow you to invoke commands without using the menus, directly from the keyboard.

olmh defines the following keyboard accelerators over the surface of the main window, except in the view area while editing a message:

Meta-I	Incorporate New Mail
Meta-C	Commit Changes
Meta-R	Rescan Folder
Meta-P	Pack Folder
Meta-S	Sort Folder
Meta-space	View Next Message
Meta-c	Mark Copy
Meta-d	Mark Deleted
Meta-f	Forward the selected or current message
Meta-m	Mark Move
Meta-n	View Next Message
Meta-p	View Previous Message
Meta-r	Reply to the selected or current message
Meta-u	Unmark

In addition to these keyboard accelerators, you can use all the standard OLIT mouseless commands to traverse and activate objects. For example, to scroll a TextEdit widget inside a ScrollingList:

Alt-Down	Scroll down a line
Alt-Up	Scroll up a line
Alt-PgDn	Scroll down a page
Alt-PgUp	Scroll up a page
Alt-Ctrl-End	Scroll to bottom
Alt-Ctrl-Home	Scroll to top

TEXT EDITING COMMANDS

All of the text editing commands are actually defined by the TextEdit widget in the OLIT Widget Set. The commands may be bound to different keys than the defaults through the OLIT key re-binding mechanisms. See the OLIT Widget Set documentation for more details.

CONFIRMATION DIALOG BOXES

Whenever you press a button that may cause you to lose some work or is otherwise dangerous, a popup dialog box will appear asking you to confirm the action. This window will contain an ''Abort'' or ''No'' button and a ''Confirm'' or ''Apply'' button. Pressing the ''No'' button cancels the operation, and pressing the ''Apply'' will proceed with the operation.

WIDGET HIERARCHY

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

The hierarchy of the main toc and view window is identical for additional toc and view windows, except that a TopLevelShell widget is inserted in the hierarchy between the application shell and the RubberTile widget.

```
Olmh olmh
        RubberTile olmh
                Control  menuBox
                        MenuButton  folderButton
                                MenuShell  menu
                                        Form  menu_form
                                                Control  pane
                                                        OblongButton  open
                                                        OblongButton  openInNew
                                                        OblongButton  create
                                                        OblongButton  delete
                                                        OblongButton  close
                        MenuButton  tocButton
```

```
        MenuShell  menu
                Form  menu_form
                        Control  pane
                                OblongButton  inc
                                OblongButton  commit
                                OblongButton  pack
                                OblongButton  sort
                                OblongButton  rescan
                                OblongButton  reverse
MenuButton  messageButton
        MenuShell  menu
                Form  menu_form
                        Control  pane
                                OblongButton  compose
                                OblongButton  next
                                OblongButton  prev
                                OblongButton  delete
                                OblongButton  move
                                OblongButton  copy
                                OblongButton  unmark
                                OblongButton  viewNew
                                OblongButton  reply
                                OblongButton  forward
                                OblongButton  useAsComp
                                OblongButton  print
MenuButton  sequenceButton
        MenuShell  menu
                Form  menu_form
                        Control  pane
                                OblongButton  pick
                                OblongButton  openSeq
                                OblongButton  addToSeq
                                OblongButton  removeFromSeq
                                OblongButton  deleteSeq
                                OblongButton  all
MenuButton  viewButton
        MenuShell  menu
                Form  menu_form
                        Control  pane
                                OblongButton  reply
                                OblongButton  forward
                                OblongButton  useAsComp
                                OblongButton  edit
                                OblongButton  save
                                OblongButton  print
```

```
StaticText folderTitlebar
Control  folders
        OblongButton  inbox
        OblongButton  drafts
        OblongButton  a_folder
        MenuButton  sub_folder
                MenuShell  menu
                        Form  menu_form
                                Control  pane
                                        OblongButton  sub_folder
                                        OblongButton sub_folder/this
                                        OblongButton sub_folder/that
                                        .
                                        .
                                        .

StaticText  tocTitlebar
ScrolledWindow  scrwin
        TextEdit  toc
StaticText  viewTitlebar
ScrolledWindow  scrwin
        TextEdit  view
```

The hierarchy of the Create Folder popup dialog box:

```
PopupWindowShell  prompt
        FooterPanel  panel
                Control  control
                        Control  upper
                                StaticText  label
                                TextField  prompt
                        Control  lower
                                OblongButton  apply
                                OblongButton  cancel
```

The hierarchy of the Notice dialog box, which reports messages from MH:

```
NoticeShell  notice
        Control  pane
                StaticText  textarea
                Control  controlarea
                        OblongButton  confirm
```

The hierarchy of the Confirmation dialog box:

```
NoticeShell  confirm
        Control  pane
                StaticText  textarea
                Control  controlarea
                        OblongButton  yes
                        OblongButton  no
```

The hierarchy of the dialog box which reports errors:

```
NoticeShell  error
        Control  pane
                StaticText  textarea
                Control  controlarea
                        OblongButton OK
```

The hierarchy of the composition window:

```
TopLevelShell  olmh
        RubberTile  olmh
                StaticText  composeTitlebar
                ScrolledWindow  scrwin
                        TextEdit  comp
                Control  compButtons
                        OblongButton  close
                        OblongButton  send
                        OblongButton  reset
                        OblongButton  compose
                        OblongButton  save
                        OblongButton  insert
```

The hierarchy of the view window:

```
TopLevelShell  olmh
        RubberTile  olmh
                StaticText  viewTitlebar
                ScrolledWindow  scrwin
                        TextEdit  view
                Control  viewButtons
                        OblongButton  close
                        OblongButton  reply
                        OblongButton  forward
                        OblongButton  useAsComp
                        OblongButton  edit
                        OblongButton  save
                        OblongButton  print
```

The hierarchy of the pick window:

Not implemented in this demo version of Olmh

APPLICATION-SPECIFIC RESOURCES

Resource instance names begin with a lower case letter but are otherwise identical to the class name.

If TocGeometry, ViewGeometry, CompGeometry, or PickGeometry are not specified, then the value of Geometry is used instead. If the resulting height is not specified (e.g., "", "=500", "+0-0"), then the default height of windows is calculated from fonts and line counts. If the width is not specified (e.g., "", "=x300", "-0+0), then half of the display width is used. If unspecified, the height of a pick window defaults to half the height of the display.

Any of these options may also be specified on the command line by using the X Toolkit 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

CheckFrequency.

CommandButtonCount
The number of command buttons to create in a button box in between the toc and the view areas of the main window. **olmh** will create these buttons with the names *button1*, *button2* and so on, in a box with the name *commandBox*. The user can specify labels and actions for the buttons in a private resource file; see the section on Actions. The default is 0.

CompGeometry
Initial geometry for windows containing compositions.

Cursor The name of the symbol used to represent the pointer. Default is ''left_ptr''.

DraftsFolder
The folder used for message drafts. Default is ''drafts''.

Geometry
Default geometry to use. Default is none.

HideBoringHeaders
If ''on'', then **olmh** will attempt to skip uninteresting header lines within messages by scrolling them off. Default is ''on''.

InitialFolder
Which folder to display on startup. May also be set with the command-line option **-initial**. Default is ''inbox''.

InitialIncFile
The file name of your incoming mail drop. **olmh** tries to construct a filename for the ''inc -file'' command, but in some installations (e.g. those using the Post Office Protocol) no file is appropriate. In this case, **InitialIncFile** should be specified as the empty string, and *inc* will be invoked without a -file argument. The default is to use the value of the environment variable **MAIL**, or if that is not set, to append the value of the environment variable **USER** to **/var/mail/**.

MailPath
The full path prefix for locating your mail folders. May also be set with the command-line option, **-path**. The default is the Path component in $HOME/.mh_profile, or ''$HOME/Mail'' if none.

MailWaitingFlag
If true, **olmh** will attempt to set an indication in its icon when new mail is waiting to be retrieved. If this option is true, then CheckNewMail is assumed to be true as well. The **-flag** command line option is a quick way to turn MailWaitingFlag on.

MakeCheckpoints
If true, **olmh** will attempt to save checkpoints of volatile information. The frequency of checkpointing is controlled by the resource **CheckFrequency**.

MhPath What directory in which to find the *MH* commands. If a command isn't found here, then the directories in the user's path are searched. Default is ''/usr/local/mh6''.

PickGeometry
Initial geometry for pick windows.

PointerColor
The foreground color of the pointer. Default is XtDefaultForeground.

PrefixWmAndIconName
Whether to prefix the window and icon name with "olmh: ". Default is true.

PrintCommand
What sh command to execute to print a message. Note that stdout and stderr must be specifically redirected! If a message or range of messages is selected for printing, the full file paths of each message file is appended to the specified print command. The default is ‘‘enscript **>**/dev/null 2**>**/dev/null’’.

ReplyInsertFilter
A shell command to be executed when the *Insert* button is activated in a composition window. The full path and filename of the source message is added to the end of the command before being passed to **sh**(1)**.** The default filter is *cat*; i.e. it inserts the entire message into the composition. Interesting filters are: *awk -e '{print "   " $0}'* or *<mh directory>/lib/mhl -form mhl.body*.

ReverseReadOrder
When true, the next message will be the message prior to the current message in the table of contents, and the previous message will be the message after the current message in the table of contents. The default is false.

SendBreakWidth
When a message is sent from **olmh**, lines longer than this value will be split into multiple lines, each of which is no longer than **SendWidth**. This value may be overridden for a single message by inserting an additional line in the message header of the form *SendBreakWidth: value*. This line will be removed from the header before the message is sent. The default is 85.

SendWidth
When a message is sent from **olmh**, lines longer than **SendBreakWidth** characters will be split into multiple lines, each of which is no longer than this value. This value may be overridden for a single message by inserting an additional line in the message header of the form *SendWidth: value*. This line will be removed from the header before the message is sent. The default is 72.

SkipCopied
Whether to skip over messages marked for copying when using ‘‘View Next Message’’ and ‘‘View Previous Message’’. Default is true.

SkipDeleted
Whether to skip over messages marked for deletion when using ‘‘View Next Message’’ and ‘‘View Previous Message’’. Default is true.

SkipMoved
Whether to skip over messages marked for moving to other folders when using ‘‘View Next Message’’ and ‘‘View Previous Message’’. Default is true.

StickyMenu
If true, when popup command menus are used, the most recently selected entry will be under the cursor when the menu pops up. Default is false. See the file *clients/olmh/Olmh.sample* for an example of how to specify resources for 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 *mhl* a lot, because it will be faster, and the extra 20 characters may not be useful.

ViewGeometry
Initial geometry for windows showing only a view of a message.

CUSTOMIZATION USING *MH*

The initial text displayed in a composition window is generated by executing the corresponding *MH* command; i.e. *comp*, *repl*, or *forw*, and therefore message components may be customized as specified for those commands. *Comp* is executed only once per invocation of **olmh** and the message template is re-used for each successive new composition.

FILES

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

SEE ALSO

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

BUGS

- Marks disappear after Incorporating new mail, although internally the messages are still marked.
- Reading mail using 'raw' MH commands or another mailer such as xmh, can cause olmh to get confused, and the 'current' message in the toc may not be the actual message displayed.
- Mouseless does not activate buttons in comp or view TopLevel shells.
- Because of the way OLIT uses the translation manager to map all events to OlAction, the action procs do not work as for xmh and have been removed from this man page.
- Printing support is minimal.
- Should handle the ''unseen'' message-sequence.
- Should determine by itself if the user hasn't used *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

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.