NAME

xmh – send and read mail with an X interface to MH

SYNOPSIS

xmh [–path *mailpath*] [–initial *foldername*] [–flag] [–*toolkitoption* ...]

DESCRIPTION

The **xmh** program provides a graphical user interface to the *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. **xmh** provides extensive mechanism for customization of the user interface.

This document introduces many aspects of the Athena Widget Set.

OPTIONS

–**path** *directory*
: This option specifies an alternate collection of mail folders in which to process mail. The directory is specified as an absolute pathname. The default mail path is the value of the Path component in the *MH* profile, which is determined by the **MH** environment variable and defaults to $HOME/.mh_profile. $HOME/Mail will be used as the path if the *MH* Path is not given in the profile.

–**initial** *folder*
: This option specifies an alternate folder which may receive new mail and is initially opened by **xmh.** The default initial folder is ''inbox''.

–**flag**
: This option will cause **xmh** to change the appearance of appropriate folder buttons and to request the window manager to change the appearance of the **xmh** icon when new mail has arrived. By default, **xmh** will change the appearance of the ''inbox'' folder button when new mail is waiting. The application-specific resource **checkNewMail** can be used to turn off this notification, and the –**flag** option will still override it.

These three options have corresponding application-specific resources, **MailPath**, **InitialFolder**, and **MailWaitingFlag**, which can be specified in a resource file.

The standard toolkit command line options are given in **X11**(7).

INSTALLATION

xmh 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 **xmh** 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 **xmh** to incorporate new mail.

Much of the user interface of **xmh** is configured in the *Xmh* application class defaults file; if this file was not installed properly a warning message will appear when **xmh** is used. **xmh** is backwards compatible with the R4 application class defaults file.

The default value of the SendBreakWidth resource has changed since R4.

BASIC SCREEN LAYOUT

xmh starts out with a single window, divided into four major areas:

- Six buttons with pull-down command menus.
- A collection of buttons, one for each top level folder. New users of *MH* will have two folders, ‘‘drafts’’ and ‘‘inbox’’.
- A listing, or Table of Contents, of the messages in the open folder. Initially, this will show the messages in ‘‘inbox’’.
- A view of one of your messages. Initially this is blank.

XMH AND THE ATHENA WIDGET SET

xmh uses the X Toolkit Intrinsics and the Athena Widget Set. Many of the features described below (scrollbars, buttonboxes, etc.) are actually part of the Athena Widget Set, and are described here only for completeness. For more information, see the Athena Widget Set documentation.

SCROLLBARS

Some parts of the main window will have a vertical area on the left containing a grey bar. This area is a *scrollbar*. They are used whenever the data in a window takes up more space than can be displayed. The grey bar indicates what portion of your data is visible. Thus, if the entire length of the area is grey, then you are looking at all your data. If only the first half is grey, then you are looking at the top half of your data. The message viewing area will have a horizontal scrollbar if the text of the message is wider than the viewing area.

You can use the pointer in the scrollbar to change what part of the data is visible. If you click with pointer button 2, the top of the grey area will move to where the pointer is, and the corresponding portion of data will be displayed. If you hold down pointer button 2, you can drag around the grey area. This makes it easy to get to the top of the data: just press with button 2, drag off the top of the scrollbar, and release.

If you click with button 1, then the data to the right of the pointer will scroll to the top of the window. If you click with pointer button 3, then the data at the top of the window will scroll down to where the pointer is.

BUTTONBOXES, BUTTONS, AND MENUS

Any area containing many words or short phrases, each enclosed in a rectangular or rounded boundary, is called a *buttonbox*. Each rectangle or rounded area is actually a button that you can press by moving the pointer onto it and pressing pointer button 1. If a given buttonbox has more buttons in it than can fit, it will be displayed with a scrollbar, so you can always scroll to the button you want.

Some buttons have pull-down menus. Pressing the pointer button while the pointer is over one of these buttons will pull down a menu. Continuing to hold the button down while moving the pointer over the menu, called dragging the pointer, will highlight each selectable item on the menu as the pointer passes over it. To select an item in the menu, release the pointer button while the item is highlighted.

ADJUSTING THE RELATIVE SIZES OF AREAS

If you're not satisfied with the sizes of the various areas of the main window, they can easily be changed. Near the right edge of the border between each region is a black box, called a *grip*. Simply point to that grip with the pointer, press a pointer button, drag up or down, and release. Exactly what happens depends on which pointer button you press.

If you drag with the pointer button 2, then only that border will move. This mode is simplest to understand, but is the least useful.

If you drag with pointer button 1, then you are adjusting the size of the window above. **xmh** will attempt to compensate by adjusting some window below it.

If you drag with pointer button 3, then you are adjusting the size of the window below. **xmh** will attempt to compensate by adjusting some window above it.

All windows have a minimum and maximum size; you will never be allowed to move a border past the point where it would make a window have an invalid size.

PROCESSING YOUR MAIL

This section will define the concepts of the selected folder, current folder, selected message(s), current message, selected sequence, and current sequence. Each **xmh** command is introduced.

For use in customization, action procedures corresponding to each command are given; these action procedures can be used to customize the user interface, particularly the keyboard accelerators and the functionality of the buttons in the optional button box created by the application resource **CommandButtonCount**.

FOLDERS AND SEQUENCES

A folder contains a collection of mail messages, or is empty. **xmh** supports folders with one level of subfolders.

The selected folder is whichever foldername appears in the bar above the folder buttons. Note that this is not necessarily the same folder that is currently being viewed. To change the selected folder, just press on the desired folder button with pointer button 1; if that folder has subfolders, select a folder from the pull-down menu.

The Table of Contents, or toc, lists the messages in the viewed folder. The title bar above the Table of Contents displays the name of the viewed folder.

The toc title bar also displays the name of the viewed sequence of messages within the viewed folder. Every folder has an implicit ''all'' sequence, which contains all the messages in the folder, and initially the toc title bar will show ''inbox:all''.

FOLDER COMMANDS

The *Folder* command menu contains commands of a global nature:

Open Folder

Display the data in the selected folder. Thus, the selected folder also becomes the viewed folder. The action procedure corresponding to this command is **XmhOpenFolder(**[*foldername*]**)**. It takes an optional argument as the name of a folder to select and open; if no folder is specified, the selected folder is opened. It may be specified as part of an event translation from a folder menu button or from a folder menu, or as a binding of a keyboard accelerator to any widget other than the folder menu buttons or the folder menus.

Open Folder in New Window
Displays the selected folder in an additional main window. Note, however, that you cannot reliably display the same folder in more than one window at a time, although **xmh** will not prevent you from trying. The corresponding action is **XmhOpenFolderInNewWindow()**.

Create Folder
Create a new folder. You will be prompted for a name for the new folder; to enter the name, move the pointer to the blank box provided and type. Subfolders are created by specifying the parent folder, a slash, and the subfolder name. For example, to create a folder named ''xmh'' which is a subfolder of an existing folder named ''clients'', type ''clients/xmh''. Click on the Okay button when finished, or just type Return; click on Cancel to cancel this operation. The action corresponding to Create Folder is **XmhCreateFolder()**.

Delete Folder
Destroy the selected folder. You will be asked to confirm this action (see CONFIRMATION WINDOWS). Destroying a folder will also destroy any subfolders of that folder. The corresponding action is **XmhDeleteFolder()**.

Close Window
Exits **xmh**, after first confirming that you won't lose any changes; or, if selected from any additional **xmh** window, simply closes that window. The corresponding action is **XmhClose()**.

HIGHLIGHTED MESSAGES, SELECTED MESSAGES, CURRENT MESSAGE

It is possible to highlight a set of adjacent messages in the area of the Table of Contents. To highlight a message, click on it with pointer button 1. To highlight a range of messages, click on the first one with pointer button 1 and on the last one with pointer button 3; or press pointer button 1, drag, and release. To extend a range of selected messages, use pointer button 3. To highlight all messages in the table of contents, click rapidly three times with pointer button 1. To cancel any selection in the table of contents, click rapidly twice.

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

The current message is indicated by a '+' next to the message number. It usually corresponds to the message currently being viewed. Upon opening a new folder, for example, the current message will be different from the viewed message. When a message is viewed, the title bar above the view will identify the message.

TABLE OF CONTENTS COMMANDS

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

Incorporate New Mail
Add any new mail received to viewed folder, and set the current message to be the first new message. This command is selectable in the menu and will execute only if the viewed folder is allowed to receive new mail. By default, only ''inbox'' is allowed to incorporate new mail. The corresponding action is **XmhIncorporateNewMail()**.

Commit Changes
Execute all deletions, moves, and copies that have been marked in this folder. The corresponding action is **XmhCommitChanges()**.

Pack Folder
Renumber the messages in this folder so they start with 1 and increment by 1. The corresponding action is **XmhPackFolder()**.

Sort Folder
Sort the messages in this folder in chronological order. (As a side effect, this may also pack the folder.) The corresponding action is **XmhSortFolder()**.

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

MESSAGE COMMANDS

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

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

View Next Message
View the first selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message after the current message. The corresponding action is **XmhViewNextMessage()**.

View Previous
View the last selected message. If no messages are highlighted, view the current message. If current message is already being viewed, view the first unmarked message before the current message. The corresponding action is **XmhViewPrevious()**.

Delete Mark the selected messages for deletion. If no messages are highlighted, mark the current message for deletion and automatically display the next unmarked message. The corresponding action is **XmhMarkDeleted()**.

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

Copy as Link Mark the selected messages to be copied into the selected folder. (If the selected folder is the same as the viewed folder, this command will just beep.) If no messages are highlighted, mark the current message to be copied. Note that messages are actually linked, not copied; editing a message copied by **xmh** will affect all copies of the message. The corresponding action is **XmhMarkCopy()**.

Unmark Remove any of the above three marks from the selected messages, or the current message, if none are highlighted. The corresponding action is **XmhUnmark()**.

View in New Create a new window containing only a view of the first selected message, or the current message, if none are highlighted. The corresponding action is **XmhViewInNewWindow()**.

Reply Create a composition window in reply to the first selected message, or the current message, if none are highlighted. The corresponding action is **XmhReply()**.

Forward Create a composition window whose body is initialized to contain an encapsulation of of the selected messages, or the current message if none are highlighted. The corresponding action is **XmhForward()**.

Use as Composition
Create a composition window whose body is initialized to be the contents of the first selected message, or the current message if none are selected. Any changes you make in the composition will be saved in a new message in the ''drafts'' folder, and will not change the original message. However, there is an exception to this rule. If the message to be used as composition was selected from the ''drafts'' folder, (see BUGS), the changes will be reflected in the original message (see COMPOSITION WINDOWS). The action procedure corresponding to this command is **XmhUseAsComposition()**.

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

SEQUENCE COMMANDS

The *Sequence* command menu contains commands pertaining to message sequences (See MESSAGE-SEQUENCES), and a list of the message-sequences defined for the currently viewed folder. The selected message-sequence is indicated by a check mark in its entry in the margin of the menu. To change the selected message-sequence, select a new message-sequence from the sequence menu.

Pick Messages Define a new message-sequence. The corresponding action is **XmhPickMessages()**.

The following menu entries will be sensitive only if the current folder has any message-sequences other than the ''all'' message-sequence.

Open Sequence Change the viewed sequence to be the same as the selected sequence. The corresponding action is **XmhOpenSequence()**.

Add to Sequence Add the selected messages to the selected sequence. The corresponding action is **XmhAddToSequence()**.

Remove from Sequence
Remove the selected messages from the selected sequence. The corresponding action is **XmhRemoveFromSequence()**.

Delete Sequence Remove the selected sequence entirely. The messages themselves are not affected; they simply are no longer grouped together to define a message-sequence. The corresponding action is **XmhDeleteSequence()**.

VIEW COMMANDS

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

Close Window When the viewed message is in a separate view window, this command will close the view, after confirming the status of any unsaved edits. The corresponding action procedure is **XmhCloseView()**.

Reply Create a composition window in reply to the viewed message. The related action procedure is **XmhViewReply()**.

Forward Create a composition window whose body is initialized contain an encapsulation of the viewed message. The corresponding action is **XmhViewForward()**.

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

the original message is edited. The action procedure corresponding to this command is **XmhViewUseAsComposition()**.

Edit Message This command enables the direct editing of the viewed message. The action procedure is **XmhEditView()**.

Save Message This command is insensitive until the message has been edited; when activated, edits will be saved to the original message in the view. The corresponding action is **XmhSaveView()**.

Print Print the viewed message. **xmh** prints by invoking the **enscript** command, but this can be customized with the application-specific resource **PrintCommand**. The corresponding action procedure is **XmhPrintView()**.

Delete Marks the viewed message for deletion. The corresponding action procedure is **XmhViewMarkDelete()**.

OPTIONS

The *Options* menu contains one entry.

Read in Reverse
When selected, a check mark appears in the margin of this menu entry. Read in Reverse will switch the meaning of the next and previous messages, and will increment to the current message marker in the opposite direction. This is useful if you want to read your messages in the order of most recent first. The option acts as a toggle; select it from the menu a second time to undo the effect. The check mark appears when the option is selected.

COMPOSITION WINDOWS

Composition windows are created by selecting **Compose Message** from the *Message* command menu, or by selecting **Reply** or **Forward** or **Use as Composition** from the *Message* or *View* command menu. These are used to compose mail messages. Aside from the normal text editing functions, there are six command buttons associated with composition windows:

Close Window Close this composition window. If changes have been made since the most recent Save or Send, you will be asked to confirm losing them. The corresponding action is **XmhCloseView()**.

Send Send this composition. The corresponding action is **XmhSend()**.

New Headers Replace the current composition with an empty message. If changes have been made since the most recent Send or Save, you will be asked to confirm losing them. The corresponding action is **XmhResetCompose()**.

Compose Message Bring up another new composition window. The corresponding action is **XmhComposeMessage()**.

Save Message Save this composition in your drafts folder. Then you can safely close the composition. At some future date, you can continue working on the composition by opening the drafts folder, selecting the message, and using the ''Use as Composition'' command. The corresponding action is **XmhSave()**.

Insert Insert a related message into the composition. If the composition window was created with a ''Reply'' command, the related message is the message being replied to, otherwise no related message is defined and this button is insensitive. The message may be filtered before being inserted; see **ReplyInsertFilter** under APPLICATION RESOURCES for more information. The corresponding action is **XmhInsert()**.

ACCELERATORS

Accelerators are shortcuts. They allow you to invoke commands without using the menus, either from the keyboard or by using the pointer.

xmh defines pointer accelerators for common actions: To select and view a message with a single click, use pointer button 2 on the message's entry in the table of contents. To select and open a folder or a sequence in a single action, make the folder or sequence selection with pointer button 2.

To mark the highlighted messages, or current message if none have been highlighted, to be moved to a folder in a single action, use pointer button 3 to select the target folder and simultaneously mark the messages. Similarly, selecting a sequence with pointer button 3 will add the highlighted or current message(s) to that sequence. In both of these operations, the selected folder or sequence and the viewed folder or sequence are not changed.

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

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

Ctrl-V	Scroll the table of contents forward
Meta-V	Scroll the table of contents backward
Ctrl-v	Scroll the view forward
Meta-v	Scroll the view backward

TEXT EDITING COMMANDS

All of the text editing commands are actually defined by the Text widget in the Athena Widget Set. The commands may be bound to different keys than the defaults described below through the X Toolkit Intrinsics key re-binding mechanisms. See the X Toolkit Intrinsics and the Athena Widget Set documentation for more details.

Whenever you are asked to enter any text, you will be using a standard text editing interface. Various control and meta keystroke combinations are bound to a somewhat Emacs-like set of commands. In addition, the pointer buttons may be used to select a portion of text or to move the insertion point in the text. Pressing pointer button 1 causes the insertion point to move to the pointer. Double-clicking button 1 selects a word, triple-clicking selects a line, quadruple-clicking selects a paragraph, and clicking rapidly five times selects everything. Any selection may be extended in either direction by using pointer button 3.

In the following, a *line* refers to one displayed row of characters in the window. A *paragraph* refers to the text between carriage returns. Text within a paragraph is broken into lines for display based on the current width of the window. When a message is sent, text is broken into lines based upon the values of the **SendBreakWidth** and **SendWidth** application-specific resources.

The following keystroke combinations are defined:

Ctrl-a	Beginning Of Line	Meta-b	Backward Word
Ctrl-b	Backward Character	Meta-f	Forward Word
Ctrl-d	Delete Next Character	Meta-i	Insert File
Ctrl-e	End Of Line	Meta-k	Kill To End Of Paragraph
Ctrl-f	Forward Character	Meta-q	Form Paragraph
Ctrl-g	Multiply Reset	Meta-v	Previous Page
Ctrl-h	Delete Previous Character	Meta-y	Insert Current Selection
Ctrl-j	Newline And Indent	Meta-z	Scroll One Line Down
Ctrl-k	Kill To End Of Line	Meta-d	Delete Next Word
Ctrl-l	Redraw Display	Meta-D	Kill Word
Ctrl-m	Newline	Meta-h	Delete Previous Word
Ctrl-n	Next Line	Meta-H	Backward Kill Word
Ctrl-o	Newline And Backup	Meta-<	Beginning Of File
Ctrl-p	Previous Line	Meta->	End Of File
Ctrl-r	Search/Replace Backward	Meta-]	Forward Paragraph
Ctrl-s	Search/Replace Forward	Meta-[	Backward Paragraph
Ctrl-t	Transpose Characters		

Ctrl-u	Multiply by 4	Meta-Delete	Delete Previous Word
Ctrl-v	Next Page	Meta-Shift Delete	Kill Previous Word
Ctrl-w	Kill Selection	Meta-Backspace	Delete Previous Word
Ctrl-y	Unkill	Meta-Shift Backspace	Kill Previous Word
Ctrl-z	Scroll One Line Up		

In addition, the pointer may be used to copy and paste text:

Button 1 Down	Start Selection
Button 1 Motion	Adjust Selection
Button 1 Up	End Selection (copy)
Button 2 Down	Insert Current Selection (paste)
Button 3 Down	Extend Current Selection
Button 3 Motion	Adjust Selection
Button 3 Up	End Selection (copy)

CONFIRMATION DIALOG BOXES

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

Some dialog boxes contain messages from *MH*. Occasionally when the message is more than one line long, not all of the text will be visible. Clicking on the message field will cause the dialog box to resize so that you can read the entire message.

MESSAGE-SEQUENCES

An *MH* message sequence is just a set of messages associated with some name. They are local to a particular folder; two different folders can have sequences with the same name. The sequence named ''all'' is predefined in every folder; it consists of the set of all messages in that folder. As many as nine sequences may be defined for each folder, including the predefined ''all'' sequence. (The sequence ''cur'' is also usually defined for every folder; it consists of only the current message. **xmh** hides ''cur'' from the user, instead placing a ''+'' by the current message. Also, **xmh** does not support *MH*'s''unseen'' sequence, so that one is also hidden from the user.)

The message sequences for a folder (including one for ''all'') are displayed in the ''Sequence'' menu, below the sequence commands. The table of contents (also known as the ''toc'') is at any one time displaying one message sequence. This is called the ''viewed sequence'', and its name will be displayed in the toc title bar after the folder name. Also, at any time one of the sequences in the menu will have a check mark next to it. This is called the ''selected sequence''. Note that the viewed sequence and the selected sequence are not necessarily the same. (This all pretty much corresponds to the way folders work.)

The **Open Sequence**, **Add to Sequence**, **Remove from Sequence**, and **Delete Sequence** commands are active only if the viewed folder contains message-sequences other than ''all'' sequence.

Note that none of the above actually affect whether a message is in the folder. Remember that a sequence is a set of messages within the folder; the above operations just affect what messages are in that set.

To create a new sequence, select the ''Pick'' menu entry. A new window will appear, with lots of places to enter text. Basically, you can describe the sequence's initial set of messages based on characteristics of the message. Thus, you can define a sequence to be all the messages that were from a particular person, or with a particular subject, and so on. You can also connect things up with boolean operators, so you can select all things from ''weissman'' with a subject containing ''xmh''.

The layout should be fairly obvious. The simplest cases are the easiest: just point to the proper field and type. If you enter in more than one field, it will only select messages which match all non-empty fields.

The more complicated cases arise when you want things that match one field or another one, but not necessarily both. That's what all the ''or'' buttons are for. If you want all things with subjects that include ''xmh'' or ''xterm'', just press the ''or'' button next to the ''Subject:'' field. Another box will appear where you can enter another subject.

If you want all things either from ''weissman'' or with subject ''xmh'', but not necessarily both, select the ''–Or–'' button. This will essentially double the size of the form. You can then enter ''weissman'' in a from: box on the top half, and ''xmh'' in a subject: box on the lower part.

If you select the ''Skip'' button, then only those messages that *don't* match the fields on that row are included.

Finally, in the bottom part of the window will appear several more boxes. One is the name of the sequence you're defining. (It defaults to the name of the selected sequence when ''Pick'' was pressed, or to ''temp'' if ''all'' was the selected sequence.) Another box defines which sequence to look through for potential members of this sequence; it defaults to the viewed sequence when ''Pick'' was pressed.

Two more boxes define a date range; only messages within that date range will be considered. These dates must be entered in RFC 822-style format: each date is of the form ''dd mmm yy hh:mm:ss zzz'', where dd is a one or two digit day of the month, mmm is the three-letter abbreviation for a month, and yy is a year. The remaining fields are optional: hh, mm, and ss specify a time of day, and zzz selects a time zone. Note that if the time is left out, it defaults to midnight; thus if you select a range of ''7 nov 86'' – ''8 nov 86'', you will only get messages from the 7th, as all messages on the 8th will have arrived after midnight.

''Date field'' specifies which field in the header to look at for this date range; it defaults to ''Date''. If the sequence you're defining already exists, you can optionally merge the old set with the new; that's what the ''Yes'' and ''No'' buttons are all about. Finally, you can ''OK'' the whole thing, or ''Cancel'' it.

In general, most people will rarely use these features. However, it's nice to occasionally use ''Pick'' to find some messages, look through them, and then hit ''Delete Sequence'' to put things back in their original state.

WIDGET HIERARCHY

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

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

```
Xmh xmh
        Paned xmh
                SimpleMenu  folderMenu
                        SmeBSB  open
                        SmeBSB  openInNew
                        SmeBSB  create
                        SmeBSB  delete
                        SmeLine  line
                        SmeBSB  close
                SimpleMenu  tocMenu
                        SmeBSB  inc
                        SmeBSB  commit
                        SmeBSB  pack
                        SmeBSB  sort
                        SmeBSB  rescan
                SimpleMenu  messageMenu
                        SmeBSB  compose
                        SmeBSB  next
                        SmeBSB  prev
                        SmeBSB  delete
                        SmeBSB  move
                        SmeBSB  copy
                        SmeBSB  unmark
                        SmeBSB  viewNew
                        SmeBSB  reply
                        SmeBSB  forward
                        SmeBSB  useAsComp
```

```
        SmeBSB  print
SimpleMenu  sequenceMenu
        SmeBSB  pick
        SmeBSB  openSeq
        SmeBSB  addToSeq
        SmeBSB  removeFromSeq
        SmeBSB  deleteSeq
        SmeLine  line
        SmeBSB  all
SimpleMenu  viewMenu
        SmeBSB  reply
        SmeBSB  forward
        SmeBSB  useAsComp
        SmeBSB  edit
        SmeBSB  save
        SmeBSB  print
SimpleMenu  optionMenu
        SmeBSB  reverse
Viewport.Core  menuBox.clip
        Box  menuBox
                MenuButton  folderButton
                MenuButton  tocButton
                MenuButton  messageButton
                MenuButton  sequenceButton
                MenuButton  viewButton
                MenuButton  optionButton
Grip  grip
Label folderTitlebar
Grip  grip
Viewport.Core  folders.clip
        Box  folders
                MenuButton  inbox
                MenuButton  drafts
                        SimpleMenu  menu
                                SmeBSB <folder_name>
                                        .
                                        .
                                        .

Grip  grip
Label  tocTitlebar
Grip  grip
Text toc
        Scrollbar  vScrollbar
Grip  grip
```

```
        Label  viewTitlebar
        Grip  grip
        Text  view
                Scrollbar  vScrollbar
                Scrollbar  hScrollbar
```

The hierarchy of the Create Folder popup dialog box:

```
TransientShell  prompt
        Dialog  dialog
                Label  label
                Text  value
                Command  okay
                Command  cancel
```

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

```
TransientShell  notice
        Dialog  dialog
                Label  label
                Text  value
                Command  confirm
```

The hierarchy of the Confirmation dialog box:

```
TransientShell  confirm
        Dialog  dialog
                Label  label
                Command  yes
                Command  no
```

The hierarchy of the dialog box which reports errors:

```
TransientShell  error
        Dialog  dialog
                Label  label
                Command  OK
```

The hierarchy of the composition window:

```
TopLevelShell  xmh
        Paned  xmh
                Label  composeTitlebar
                Text  comp
                Viewport.Core  compButtons.clip
```

```
                        Box  compButtons
                                Command  close
                                Command  send
                                Command  reset
                                Command  compose
                                Command  save
                                Command  insert
```

The hierarchy of the view window:

```
        TopLevelShell  xmh
                Paned  xmh
                        Label  viewTitlebar
                        Text  view
                        Viewport.Core  viewButtons.clip
                                Box  viewButtons
                                        Command  close
                                        Command  reply
                                        Command  forward
                                        Command  useAsComp
                                        Command  edit
                                        Command  save
                                        Command  print
                                        Command  delete
```

The hierarchy of the pick window:
(Unnamed widgets have no name.)

```
        TopLevelShell  xmh
                Paned  xmh
                        Label  pickTitlebar
                        Viewport.Core  pick.clip
                                Form  form
                                        Form  groupform
```

The first 6 rows of the pick window have identical structure:

```
                                                Form  rowform
                                                        Toggle
                                                        Toggle
                                                        Label
                                                        Text
                                                        Command

                                                Form  rowform
                                                        Toggle
                                                        Toggle
```

```
                                    Text
                                    Text
                                    Command
                                Form  rowform
                                    Command
                    Viewport.core  pick.clip
                        Form  form
                            From  groupform
                                Form  rowform
                                    Label
                                    Text
                                    Label
                                    Text
                                Form  rowform
                                    Label
                                    Text
                                    Label
                                    Text
                                    Label
                                    Text
                                Form  rowform
                                    Label
                                    Toggle
                                    Toggle
                                Form  rowform
                                    Command
                                    Command
```

APPLICATION-SPECIFIC RESOURCES

The application class name is **Xmh**. Application-specific resources are listed below by name. Application-specific resource class names always begin with an upper case character, but unless noted, are otherwise identical to the instance names given below.

Any of these options may also be specified on the command line by using the X Toolkit Intrinsics resource specification mechanism. Thus, to run **xmh** showing all message headers,
% xmh –xrm ’∗HideBoringHeaders:off’

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

The following resources are defined:

banner A short string that is the default label of the folder, Table of Contents, and view. The default is "xmh MIT X Consortium R5".

blockEventsOnBusy
Whether to disallow user input and show a busy cursor while **xmh** is busy processing a command. Default is true.

busyCursor
The name of the symbol used to represent the position of the pointer, displayed if **blockEventsOnBusy** is true, when **xmh** is processing a time-consuming command. The default is "watch".

busyPointerColor
The foreground color of the busy cursor. Default is XtDefaultForeground.

checkFrequency
How often to check for new mail, make checkpoints, and rescan the Table of Contents, in minutes. If **checkNewMail** is true, **xmh** checks to see if you have new mail each interval. If **makeCheckpoints** is true, checkpoints are made every fifth interval. Also every fifth interval, the Table of Contents is checked for inconsistencies with the file system, and rescanned if out of date. To prevent all of these checks from occurring, set **CheckFrequency** to 0. The default is 1. This resource is retained for backward compatibility with user resource files; see also **checkpointInterval**, **mailInterval**, and **rescanInterval**.

checkNewMail
If true, **xmh** will check at regular intervals to see if new mail has arrived for any of the top level folders and any opened subfolders. A visual indication will be given if new mail is waiting to be incorporated into a top level folder. Default is true. The interval can be adjusted with **mailInterval**.

checkpointInterval (class **Interval**)
Specifies in minutes how often to make checkpoints of volatile state, if **makeCheckpoints** is true. The default is 5 times the value of **checkFrequency**.

checkpointNameFormat
Specifies how checkpointed files are to be named. The value of this resource will be used to compose a file name by inserting the message number as a string in place of the required single occurance of '%d'. If the value of the resource is the empty string, or if no '%d' occurs in the string, or if "%d" is the value of the resource, the default will be used instead. The default is "%d.CKP". Checkpointing is done in the folder of origin unless an absolute pathname is given. **xmh** does not assist the user in recovering checkpoints, nor does it provide for removal of the checkpoint files.

commandButtonCount
The number of command buttons to create in a button box in between the toc and the view areas of the main window. **xmh** will create these buttons with the names *button1*, *button2* and so on, in a box with the name *commandBox*. The default is 0. **xmh** users can specify labels and actions for the buttons in a private

resource file; see the section ACTIONS AND INTERFACE CUSTOMIZATION.

compGeometry
Initial geometry for windows containing compositions.

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

debug Whether or not to print information to stderr as **xmh** runs. Default is false.

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

geometry
Default geometry to use. Default is none.

hideBoringHeaders
If ''on'', then **xmh** will attempt to skip uninteresting header lines within messages by scrolling them off the top of the view. Default is ''on''.

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

initialIncFile
The absolute path name of your incoming mail drop file. In some installations, for example those using the Post Office Protocol, no file is appropriate. In this case, **initialIncFile** should not be specified, or may be specified as the empty string, and *inc* will be invoked without a −file argument. By default, this resource has no value. This resource is ignored if **xmh** finds an *.xmhcheck* file; see the section on multiple mail drops.

mailInterval (class **Interval)**
Specifies the interval in minutes at which the mail should be checked, if **mailWaitingFlag** or **checkNewMail** is true. The default is the value of **checkFrequency**.

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

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

makeCheckpoints
If true, **xmh** will attempt to save checkpoints of volatile edits. The default is false. The frequency of checkpointing is controlled by the resource **checkpointInterval**. For the location of checkpointing, see **checkpointNameFormat**.

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

newMailBitmap (class **NewMailBitmap**)
The bitmap to show in the folder button when a folder has new mail. The default is ''black6''.

newMailIconBitmap (class **NewMailBitmap**)
The bitmap suggested to the window manager for the icon when any folder has new mail. The default is ''flagup''.

noMailBitmap (class **NoMailBitmap)**
The bitmap to show in the folder button when a folder has no new mail. The default is ''box6''.

noMailIconBitmap (class **NoMailBitmap)**
The bitmap suggested to the window manager for the icon when no folders have new mail. The default is ''flagdown''.

pickGeometry
Initial geometry for pick windows.

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

prefixWmAndIconName
Whether to prefix the window and icon name with "xmh: ". Default is true.

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

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

rescanInterval (class **Interval**)
How often to check the Table of Contents of currently viewed folders and of folders with messages currently being viewed, and to update the Table of Contents if **xmh** sees inconsistencies with the file system in these folders. The default is 5 times the value of **checkFrequency**.

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

sendBreakWidth
When a message is sent from **xmh**, lines longer than this value will be split into multiple lines, each of which is no longer than **SendWidth**. This value may be overridden for a single message by inserting an additional line in the message header of the form *SendBreakWidth: value*. This line will be removed from the header before the message is sent. The default is 2000 (to allow for sending mail containing source patches).

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

showOnInc
Whether to automatically show the current message after incorporating new mail. Default is true.

skipCopied
Whether to skip over messages marked for copying when using ''View Next Message'' and ''View Previous Message''. Default is true.

skipDeleted
Whether to skip over messages marked for deletion when using ''View Next Message'' and ''View Previous Message''. Default is true.

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

stickyMenu
If true, when popup command menus are used, the most recently selected entry will be under the cursor when the menu pops up. Default is false. See the file *clients/xmh/Xmh.sample* for an example of how to specify resources for popup command menus.

tempDir
Directory for **xmh** to store temporary files. For privacy, a user might want to change this to a private directory. Default is ''/tmp''.

tocGeometry
Initial geometry for main **xmh** toc and view windows.

tocPercentage
The percentage of the main window that is used to display the Table of Contents. Default is 33.

tocWidth
How many characters to generate for each message in a folder's table of contents. Default is 100. Use less if the geometry of the main **xmh** window results in the listing being clipped at the right hand boundary, or if you plan to use *mhl*

a lot, because it will be faster, and the extra characters may not be useful.

viewGeometry
Initial geometry for windows showing a view of a message.

MULTIPLE MAIL DROPS

Users may need to incorporate mail from multiple spool files or mail drops. If incoming mail is forwarded to the *MH slocal* program, it can be sorted as specified by the user into multiple incoming mail drops. Refer to the *MH* man page for *slocal* to learn how to specify fowarding and the automatic sorting of incoming mail in a *.maildelivery* file.

To inform **xmh** about the various mail drops, create a file in your home directory called *.xmhcheck*. In this file, a mapping between existing folder names and mail drops is created by giving a folder name followed by the absolute pathname of the mail drop site, with some white space separating them, one mapping per line. **xmh** will read this file whether or not resources are set for notification of new mail arrival, and will allow incorporation of new mail into any folder with a mail drop. **xmh** will invoke *inc* with the *−file* argument, and if **xmh** has been requested to check for new mail, it will check directly, instead of using *msgchk*.

An example of *.xmhcheck* file format, for the folders ''inbox'' and ''xpert'':

```
inbox    /var/mail/converse
xpert    /users/converse/maildrops/xpert
```

ACTIONS AND INTERFACE CUSTOMIZATION

Because **xmh** provides action procedures which correspond to command functionality and installs accelerators, users can customize accelerators and new button functionality in a private resource file. For examples of specifying customized resources, see the file *mit/clients/xmh/Xmh.sample*. To understand the syntax, see the Appendix of the *X Toolkit Intrinsics* specification on *Translation Table Syntax*, and any general explanation of using and specifying *X* resources. Unpredictable results can occur if actions are bound to events or widgets for which they were not designed.

Here's an example of how to bind actions to your own **xmh** buttons, and how to redefine the default accelerators so that the Meta key is not required, in case you don't have access to the sample file mentioned above.

```
! To create buttons in the middle of the main window and give them semantics:

Xmh*CommandButtonCount:               5

Xmh*commandBox.button1.label:         Inc
Xmh*commandBox.button1.translations: #override\
        <Btn1Down>,<Btn1Up>: XmhIncorporateNewMail() unset()

Xmh*commandBox.button2.label:         Compose
Xmh*commandBox.button2.translations: #override\
        <Btn1Down>,<Btn1Up>: XmhComposeMessage() unset()
```

```
Xmh*commandBox.button3.label:         Next
Xmh*commandBox.button3.translations: #override\
        <Btn1Down>,<Btn1Up>: XmhViewNextMessage() unset()

Xmh*commandBox.button4.label:         Delete
Xmh*commandBox.button4.translations: #override\
        <Btn1Down>,<Btn1Up>: XmhMarkDelete() unset()

Xmh*commandBox.button5.label:         Commit
Xmh*commandBox.button5.translations: #override\
        <Btn1Down>,<Btn1Up>: XmhCommitChanges() unset()

! To redefine the accelerator bindings to exclude modifier keys,
! and add your own keyboard accelerator for Compose Message:

Xmh*tocMenu.accelerators: #override\n\
        !:<Key>I:       XmhIncorporateNewMail()\n\
        !:<Key>C:       XmhCommitChanges()\n\
        !:<Key>R:       XmhForceRescan()\n\
        !:<Key>P:       XmhPackFolder()\n\
        !:<Key>S:       XmhSortFolder()\n
Xmh*messageMenu.accelerators: #override\n\
        !:<Key>E:       XmhComposeMessage()\n\
        !<Key>space:    XmhViewNextMessage()\n\
        !:<Key>c:       XmhMarkCopy()\n\
        !:<Key>d:       XmhMarkDelete()\n\
        !:<Key>f:       XmhForward()\n\
        !:<Key>m:       XmhMarkMove()\n\
        !:<Key>n:       XmhViewNextMessage()\n\
        !:<Key>p:       XmhViewPreviousMessage()\n\
        !:<Key>r:       XmhReply()\n\
        !:<Key>u:       XmhUnmark()\n
```

xmh provides action procedures which correspond to entries in the command menus; these are given in the sections describing menu commmands, not here. In addition to the actions corresponding to commands in the menus, these action routines are defined:

XmhPushFolder([*foldername*, ...]**)**
: This action pushes each of its argument(s) onto a stack of foldernames. If no arguments are given, the selected folder is pushed onto the stack.

XmhPopFolder()
: This action pops one foldername from the stack and sets the selected folder.

XmhPopupFolderMenu()
: This action should always be taken when the user selects a folder button. A folder button represents a folder and zero or more subfolders. The menu of subfolders is built upon the first reference, by this routine. If there are no

subfolders, this routine will mark the folder as having no subfolders, and no menu will be built. In that case the menu button emulates a toggle button. When subfolders exist, the menu will popup, using the menu button action PopupMenu().

XmhSetCurrentFolder()
This action allows menu buttons to emulate toggle buttons in the function of selecting a folder. This action is for menu button widgets only, and sets the selected folder.

XmhLeaveFolderButton()
This action ensures that the menu button behaves properly when the user moves the pointer out of the menu button window.

XmhPushSequence([*sequencename*, ...]**)**
This action pushes each of its arguments onto the stack of sequence names. If no arguments are given, the selected sequence is pushed onto the stack.

XmhPopSequence()
This action pops one sequence name from the stack of sequence names, which then becomes the selected sequence.

XmhPromptOkayAction()
This action is equivalent to pressing the okay button in the Create Folder popup.

XmhReloadSeqLists()
This action rescans the contents of the public *MH* sequences for the currently opened folder and updates the sequence menu if necessary.

XmhShellCommand(*parameter* [, *parameter*]**)**
At least one parameter must be specified. The parameters will be concatenated with a space character separator, into a single string, and the list of selected messsages, or if no messages are selected, the current message, will be appended to the string of parameters. The string will be executed as a shell command. The messages are always given as absolute pathnames. It is an error to cause this action to execute when there are no selected messages and no current message.

XmhCheckForNewMail()
This action will check all mail drops known to xmh. If no mail drops have been specified by the user either through the *.xmhcheck* file or by the **initialIncFile** resource, the *MH* command *msgchk* is used to check for new mail, otherwise, **xmh** checks directly.

XmhWMProtocols([wm_delete_window] [wm_save_yourself])
This action is responsible for participation in window manager communication protocols. It responds to delete window and save yourself messages. The user can cause **xmh** to respond to one or both of these protocols, exactly as if the window manager had made the request, by invoking the action with the appropriate parameters. The action is insensitive to the case of the string parameters. If the event received is a ClientMessage event and parameters are

present, at least one of the parameters must correspond to the protocol requested by the event for the request to be honored by **xmh**.

CUSTOMIZATION USING *MH*

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

xmh uses *MH* commands, including *inc*, *msgchk*, *comp*, *send*, *repl*, *forw*, *refile*, *rmm*, *pick*, *pack*, *sort*, and *scan*. Some flags for these commands can be specified in the *MH* profile; **xmh** may override them. The application resource **debug** can be set to true to see how **xmh** uses *MH* commands.

ENVIRONMENT

HOME - users's home directory
MH - to get the location of the *MH* profile file

FILES

~/.mh_profile - *MH* profile, used if the MH environment variable is not set
~/Mail - directory of folders, used if the *MH* profile cannot be found
~/.xmhcheck - optional, for multiple mail drops in cooperation with *slocal*.
/usr/local/mh6 - *MH* commands, as a last resort, see **mhPath**.
~/Mail/<folder>/.xmhcache - *scan* output in each folder
~/Mail/<folder>/.mh_sequences - sequence definitions, in each folder
/tmp - temporary files, see **tempDir**.

SEE ALSO

X11(7), **xrdb**(1), X Toolkit Intrinsics, Athena Widget Set
At least one book has been published about *MH* and **xmh**.

BUGS

- When the user closes a window, all windows which are transient for that window should also be closed by **xmh.**
- When **XmhUseAsComposition** and **XmhViewUseAsComposition** operate on messages in the **DraftsFolder**, **xmh** disallows editing of the composition if the same message is also being viewed in another window.
- Occasionally after committing changes, the table of contents will appear to be completely blank when there are actually messages present. When this happens, refreshing the display, or typing Control-L in the table of contents, will often cause the correct listing to appear. If this doesn't work, force a rescan of the folder.
- Should recognize and use the ''unseen'' message-sequence.
- Should determine by itself if the user hasn't used *MH* before, and offer to create the .mh_profile, instead of hanging on inc.
- A few commands are missing (rename folder, resend message).
- WM_DELETE_WINDOW protocol doesn't work right when requesting deletion of the first toc and view, while trying to keep other **xmh** windows around.
- Doesn't support annotations when replying to messages.
- Doesn't allow folders to be shared without write permission.

- Doesn’t recognize private sequences.
- *MH* will report that the *.mh_sequences* file is poorly formatted if any sequence definition in a particular folder contains more than *BUFSIZ* characters. **xmh** tries to capture these messages and display them when they occur, but it cannot correct the problem.

COPYRIGHT

AUTHOR

Terry Weissman, formerly of Digital Western Research Laboratory
Donna Converse, MIT X Consortium