

# Applixware Macro Editor ELF Reference

COPYRIGHT NOTICE ON THE VERSION 6.0 SOFTWARE  
©1990 - 2010 Vistasource, Inc. All Rights Reserved.

Vistasource, Inc. prepared the information contained in this document for use by Vistasource personnel, customers, and prospects. Vistasource reserves the right to change the information in this document without prior notice. The contents herein should not be construed as a representation or warranty by Vistasource. Vistasource assumes no responsibility for any errors that may appear in this document.

The Proximity Thesauri ®  
©1985 Merriam-Webster Inc.  
©1988 Williams Collins Sons & Co. Ltd.  
©1989 Van Dale Lexicografie bv. ©1989 Nathan. ©1989 Kruger.  
©1989 Zanichelli. ©1989 International Data Education a s.  
©1989 C.A. Stromber A B. ©1989 Espasa-Calpe.  
©1983-1996. Proximity Technology, Inc.  
All Rights Reserved.

The Proximity Linguibase And Hyphenation Systems®  
©1983 Merriam-Webster Inc.  
©1984, 1985, 1986, 1988, 1990 Williams Collins Sons & Co. Ltd.  
©1987, 1989 Van Dale Lexicografie bv. ©1988 Munksgaard International Publishers Ltd.  
©1988, 1989 International Data Education a s.  
©1983-1996 Proximity Technology, Inc.  
All Rights Reserved

The Applixware Graphics Filter Pack contains elements of the Generator Metafile Development Libraries (MDL/G)  
©1988-1996 Henderson Software, Inc.  
All Rights Reserved

### RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraphs (c) (1) (ii) of SFARS 252.277-7013, or in FAR 52.227-19, as applicable.

Hardware and software products mentioned herein are used for identification purposes only and may be trademarks of their respective companies.

Applixware is a registered trademark of Vistasource, Inc. Applixware, Applixware Real Time, Applixware Data, and Applixware Builder are trademarks of Vistasource, Inc.

**This manual was produced using Applixware.**

Printed: June 2010

---

## ME\_APPLICATION\_DLG@

---

Creates a new Macro Editor window

**Format** taskID =ME\_APPLICATION\_DLG@([menubarID],  
[,windowlessFlag][, hooklessFlag ])

**Arguments** menubarID The number of a menubar to be associated with this window. This number should be a number between 400 and 499.

windowlessFlag

Boolean value where TRUE indicates that no window will be displayed. FALSE is the default.

hooklessFlag A Boolean value where TRUE indicates that no hook macro is run at startup time. If the value is FALSE, the macro editor runs the hook macro configured in the macro editor preferences.

winTitle An optional name to be used as the Macro Editor Window's title.

**Description** Creates a new Macro Editor window. This window can use the default menubar or it can use the menubar associated with menubar\_id. The task id of the newly created Macro Editor window is returned.

The optional menubarID parameters lets you load a menubar according to the task that runs in the Macro Editor window. You could even display several versions at the same time by opening several windows, each with separate menubars. To display a custom menubar, you have to first load it into memory using **SET SELECTIONS@**.

The windowlessFlag parameter lets you perform automated tasks "in the background," without displaying the Macro Editor application window. Using this option, two applications can work simultaneously without interrupting each other and without calling two separate axmain processes. This windowless Macro Editor application becomes a "child" of the window from which it was invoked.

Performing non-interactive tasks without a window conserves computing resources as the window is not displayed.

To perform a windowless task:

- You must suppress all messages that are displayed. Otherwise, the task hangs when it tries to display them. Use the following macro to suppress information displayed by INFO\_MESSAGE@:

**SUPPRESS INFO MESSAGES@**

To suppress error messages use the following macro:

**SUPPRESS ERROR MESSAGES@**

- You cannot include any prompts using PROMPT@.
- Be sure to explicitly exit any windowless Macro Editor applications after they complete. Otherwise, the tasks remain in memory until you log out.
- It is best to make a call to **SELECT\_WINDOW@** when exiting the windowless application. In this way, you guarantee that the exit command is invoked against the right window.
- Before testing a newly written macro that calls a windowless application, it is best to first test the macro with all windows displaying in the foreground.

The hooklessFlag parameter determines whether a hook macro is run when the Macro editor starts. A hook macro is configured through the macro editor preferences dialog, and runs whenever you start the macro editor or open a macro file.

If you configure a hook macro to run at startup time, and you set hooklessFlag to NULL, the hook macro runs. If you configure a hook macro to run at startup time, and you set hooklessFlag to TRUE, the hook macro does not run. This is an optional parameter.

ME\_APPLICATION\_DLG@ is called by the \* ® Macro Editor menu option.

---

## ME\_COMPILE@

---

Compiles the current macro file

**Format** ME\_COMPILE@()

**Description** Compiles the current macro document. If syntax errors occur when compiling, error messages are placed in the macro document where the errors occur and ME\_COMPILE@ is terminated.

**See also** **ME\_COMP\_SAVE\_INSTALL@**

---

## ME\_COMPILE\_FILE@

---

Compiles a named file

**Format** ME\_COMPILE\_FILE@(filename)

**Arguments** filename      The name of the file being compiled

**Description** Compiles all the macros in filename. If syntax errors occur when compiling, an error message is displayed. However, unlike **ME\_COMPILE@**, this macro does not place the error messages in the file.

---

## ME\_COMP\_SAVE\_INSTALL@

---

Compiles, saves, and installs a document's macros

**Format** ME\_COMP\_SAVE\_INSTALL@()

**Description** Compiles and saves the current macro document. If compilation was successful, the macros are installed within the ELF runtime environment. If any compilation errors occur, error messages are placed in the macro document where the errors occur and the execution of ME\_COMP\_SAVE\_INSTALL@ is terminated.

ME\_COMPILE\_SAVE\_INSTALL@ is called by the File ® Compile and Save menu option in the Macro Editor.

**See also** [ME\\_COMPILE@](#)

[ME\\_SAVE@](#)

---

## ME\_COPY@

---

Copies selected text to the clipboard

**Format** ME\_COPY@()

**Description** Copies selected text from the current Macro Editor document and places this information in the clipboard. If no text is selected, an error occurs. ME\_COPY@ is called by Edit ® Copy.

**See also** [ME\\_CUT@](#)

[ME\\_DELETE@](#)

[ME\\_PASTE@](#)

---

## ME\_CUT@

---

Removes selected text, placing it on the clipboard

**Format** ME\_CUT@()

**Description** Cuts selected text from the current Macro Editor document and places the text in the clipboard. If no text is selected, an error occurs. ME\_CUT@ is called by Edit ® Cut.

See also [ME\\_COPY@](#)  
[ME\\_DELETE@](#)  
[ME\\_PASTE@](#)

---

## ME\_DELETE@

---

Deletes selected text

**Format** ME\_DELETE@()

**Description** Deletes all selected text in the current Macro Editor document. Deleted text is not saved in the clipboard. ME\_DELETE@ is called by Edit ® Delete.

See also [ME\\_CUT@](#)  
[ME\\_DELETE\\_ERROR\\_LINES@](#)  
[ME\\_DELETE\\_LINE@](#)  
[ME\\_DELETE\\_TO\\_EOL@](#)  
[ME\\_DELETE\\_WORD@](#)

---

## ME\_DELETE\_DOC@

---

Deletes the current Macro Editor document

**Format** ME\_DELETE\_DOC@()

**Description** Deletes the current Macro Editor document and closes the Macro Editor window. Any changes made to the document that were not saved are lost.

---

## ME\_DELETE\_ERROR\_LINES@

---

Removes all error messages

**Format** ME\_DELETE\_ERROR\_LINES@()

**Description** Removes all error message lines that were inserted into the file during a compilation. ME\_DELETE\_ERROR\_LINES@ is only relevant after an attempt has been made to compile the document and errors are found. ME\_DELETE\_ERROR\_LINES@ is called by the View ® Delete Error Messages menu option in the Macro Editor.

See also [ME\\_DELETE@](#)

---

## ME\_DELETE\_LINE@

---

Deletes the current line

**Format** ME\_DELETE\_LINE@()

**Description** Deletes the current line in the current Macro Editor document.

See also [ME\\_DELETE@](#)

---

## ME\_DELETE\_TO\_EOL@

---

Deletes from the cursor position to the end of the line

**Format** ME\_DELETE\_TO\_EOL@()

**Description** Deletes from the current cursor position to the end of the line.

See also [ME\\_DELETE@](#)

---

## ME\_DELETE\_WORD@

---

Deletes the word at the current cursor position

**Format** ME\_DELETE\_WORD@()

**Description** If the cursor is in a space, the first word following the space is deleted. If there is a selection, all words containing selected characters are deleted. In either case, the trailing space following the last word is also deleted. ME\_DELETE\_WORD@ is called by Keys® Delete Word.

See also [ME\\_DELETE@](#)

---

## ME\_DOC\_IS\_OPEN@

---

Determines if a Macro Editor document is open

**Format** flag =ME\_DOC\_IS\_OPEN@(filename)

**Arguments** filename     The name of the passed document.

**Description** Returns TRUE if the passed Macro Editor document is currently open. If it is not open, FALSE is returned.

---

### ME\_FIND\_NEXT@

---

Finds next occurrence of data

**Format** ME\_FIND\_NEXT@()

**Description** Beginning at the current cursor position, searches forward through the current macro document for the next occurrence of a string. This string was previously specified using the Find ® Find & Replace command. If another occurrence exists, it is highlighted.

**See also** [ME\\_FIND\\_PREV@](#)

---

### ME\_FIND\_PREV@

---

Finds previous occurrence of data

**Format** ME\_FIND\_PREV@()

**Description** Beginning at the current cursor position, searches backward through the current macro document for the previous occurrence of a string. This string was previously specified using the Find ® Find & Replace command. If another occurrence exists, it is highlighted.

**See also** [ME\\_FIND\\_NEXT@](#)

---

### ME\_GET\_DOC\_INFO@

---

Returns information on the current document

**Format** info = ME\_GET\_DOC\_INFO@()

**Description** Returns the following information on the current Words document. The FORMAT template for the array information is named doc\_format\_. The header file containing the FORMAT template is fileinf\_.am. Here is the format's definition:

```
format doc_format_,  
       name,          ' The full path name of the current document.
```

docid,	' Not important: The document's unique ID number.
on_disk,	' Indicates if the document has been saved or read using the file name specified by name. Returns TRUE if the file has been saved or read. Returns FALSE if it hasn't.
save_mode,	' The mode in which the document has been saved. save_mode can be one of the following: '0 Not supported: binary format '1 Machine-independent (portable) format
grp_access,	' The permission setting for the file for members of the same group. grp_access can be one of the following: '0 No read or write permissions for the file '1 Read permission for the file. '2 Read and write permission for the file
all_access,	' One of the following file permissions for the users: '0 No read or write permissions for the file '1 Read permission for the file. '2 Read and write permission for the file.
writeable,	' Indicates whether this user has write access. TRUE indicates that this user can write this file; the value FALSE indicates that the user cannot write this file.

---

## ME\_GOTO\_BEGINNING@

---

Moves cursor to the beginning of the document

**Format** ME\_GOTO\_BEGINNING@()

**Description** Places the cursor before the first character in the current Words document and updates the Words display to show the first page in the document.

**See also** [ME\\_GOTO\\_END@](#)

[ME\\_GOTO\\_LINE@](#)

---

## ME\_GOTO\_BEGIN\_LINE@

---

Moves the cursor to the beginning of the line

**Format** ME\_GOTO\_BEGIN\_LINE@()

**Description** Moves the cursor so that it is placed before the first character in the current line.

**See also** [ME\\_GOTO\\_END\\_LINE@](#)

---

## ME\_GOTO\_END@

---

Moves the cursor to the end of the document

**Format** ME\_GOTO\_END@()

**Description** Places the cursor after the last character in the current Macro Editor document and updates the display to show this position.

**See also** [ME\\_GOTO\\_BEGINNING@](#)  
[ME\\_GOTO\\_LINE@](#)

---

## ME\_GOTO\_END\_LINE@

---

Moves the cursor to the end of the current line

**Format** ME\_GOTO\_END\_LINE@()

**Description** Moves the cursor so that it is immediately after the last character on the current line. Because the Macro Editor is not line-oriented, this position really does not exist; this position is actually immediately before the first character on the *next* line, which is the position usually shown.

For example, if a line break appears between two words, the space character between the two words is eliminated from the display, which means the character immediately in front of the cursor does not exist.

**See also** [ME\\_GOTO\\_BEGIN\\_LINE@](#)

---

## ME\_GOTO\_LINE@

---

Moves to the specified line

**Format** ME\_GOTO\_LINE@(lineNum)

**Arguments** lineNum      A line in the Macro Editor document.

**Description** Moves the cursor to the beginning of the line you specify. Lines are numbered beginning at line number 1.

If lineNum is larger than the number of lines in the document, the cursor is placed at the end of the document.

A Macro Editor line includes any text that auto-wraps. For example, if one line auto-wraps onto four lines, the Macro Editor considers this to be 1 line, not 4 lines.

See also [ME\\_GOTO\\_BEGINNING@](#)  
[ME\\_GOTO\\_END@](#)

---

## ME\_GOTO\_NEXT\_ERROR@

---

Moves the cursor to the next error message

**Format** ME\_GOTO\_NEXT\_ERROR@()

**Description** ME\_GOTO\_NEXT\_ERROR@ searches forward from the current cursor location for the next error message in the current macro document. The cursor is placed at the beginning of the next error message.

ME\_GOTO\_NEXT\_ERROR@ is only relevant after an attempt has been made to compile the document. If there are no more error messages in the document, the cursor is not moved and the system beeps.

ME\_GOTO\_NEXT\_ERROR@ is called by the Find ® Next Error menu option in the Macro Editor.

See also [ME\\_GOTO\\_PREV\\_ERROR@](#)

---

## ME\_GOTO\_PREV\_ERROR@

---

Moves the cursor to the previous error message

**Format** ME\_GOTO\_PREV\_ERROR@()

**Description** ME\_GOTO\_PREV\_ERROR@ searches backwards from the current cursor location for the previous error message in the current macro document. The cursor is placed at the beginning of the previous error message.

ME\_GOTO\_PREV\_ERROR@ is only relevant after an attempt has been made to compile the document. If there are no more error messages in the document, the cursor is not moved and the system beeps.

ME\_GOTO\_PREV\_ERROR@ is called by the Find ® Previous Error menu option in the Macro Editor.

See also [ME\\_GOTO\\_NEXT\\_ERROR@](#)

---

## ME\_IMPORT\_ASCII@

---

Displays an ASCII file in a Macro Editor window

**Format** ME\_IMPORT\_ASCII@(filename)

**Arguments** filename      The full path name of the ASCII file to import.

**Description** Imports an ASCII file into a new Macro Editor window. Each line in the ASCII file is imported as a separate line in the macro document.

---

## ME\_IN\_DEBUG\_MODE@

---

Indicates whether the current Macro Editor window is in debug mode

**Format** flag = ME\_IN\_DEBUG\_MODE@()

**Description** Returns TRUE if the current Macro Editor window is in debug mode. It returns FALSE if the window is not in debug mode.

**See also** [ME\\_TOGGLE\\_DEBUGGER@](#)

---

## ME\_INITIALIZE\_DOC@

---

Sets a document's language

**Format** ME\_INITIALIZE\_DOC@( )

**Description** Sets the language mode for a document that is presumed to have just been opened. This macro should be called by all macros that open and create new Applixware documents.

It also resets the document's *modified* flag so that if a document is closed at the point, the user will not be asked to save the file.

---

## ME\_INSERT\_ERROR@

---

Inserts the error message before a bead position

**Format** ME\_INSERT\_ERROR@(errString, beadNum)

**Arguments** errString The text of the error message.  
beadNum The bead position at which the error message is inserted.

**Description** Inserts an error message at the indicated bead number. The character attributes of the error message are predefined.  
The contents of the error message are created with a bead type named *Error*. The contents of this bead cannot be edited and are never written to disk.

---

### ME\_LOAD\_FILE@

---

Loads a file in the current Macro Editor window

**Format** ME\_LOAD\_FILE@(filename[, noHook])

**Arguments** filename The name of the macro file being loaded.  
noHook A Boolean value which if set to TRUE indicates that Words should not execute the file's hook macro.

**Description** Loads the specified file into the current Macro Editor window. If you have not saved changes to the current document, the changes are lost.  
If the document specified by filename does not exist, a blank macro document is displayed in the current window and the window is given the name specified by filename.

---

### ME\_MAIL@

---

Mails the current document

**Format** ME\_MAIL@()

**Description** Mails the current Macro editor document. If there are linked objects, these linked objects are also sent.

---

### ME\_NEW@

---

Opens a new Macro Editor document

**Format** ME\_NEW@()

**Description** Opens a new, blank Macro Editor document. A save on this document forces the File ® Save As menu option. ME\_NEW@ is called by the File ® New menu option.

---

## ME\_NEW\_WINDOW@

---

Opens a new, empty Macro Editor window

**Format** pid = ME\_NEW\_WINDOW@(id, quietFlag, readonlyFlag, width, height[, windowTitle[, iconTitle[, iconID ] ]])

**Arguments**

id	The menu id, use the standard Words menus or a custom menu.
quiet	If set to TRUE, the new window runs in <i>windowless</i> mode. See the <a href="#">applix -quiet</a> command line option. If set to FALSE, the window runs normally.
readOnly	If set to TRUE, the new window runs in Read-Only mode. If set to FALSE, the window runs normally.
width	A non-zero value (in mils, which are set 1000 to the inch) that overrides the standard window width.
height	A non-zero value (in mils, which are set 1000 to the inch) that overrides the standard window height.
windowTitle	An optional window title. This title will not be changed when new files are read into the window.
iconTitle	An optional title to be used when the window is minimized.
iconId	An optional icon id; this icon will be used when the window is minimized.

**Description** Opens a new Words window.  
If a file hook is set, it is run with a value of 1.

---

## ME\_PASTE@

---

Pastes the contents of the clipboard

**Format** ME\_PASTE@()

**Description** Pastes the contents of the clipboard at the current cursor position in the current Words document. ME\_PASTE@ is called by Edit ® Paste.

**See also** [ME\\_COPY@](#)  
[ME\\_CUT@](#)

---

## ME\_POINTER\_TYPE@

---

Sets the text cursor's shape

**Format** ME\_POINTER\_TYPE@(pointerStyle)

**Arguments** pointerStyle The name of the new pointer style.

**Description** Sets the new text pointer shape. To obtain a list of available shape names, use [GET\\_CURSOR\\_LIST@](#).

---

## ME\_PRINT@

---

Prints the current Macro Editor document

**Format** ME\_PRINT@(printer, colorFlag, copies, bannerFlag, allPagesFlag, startPage, endPage, FALSE, mergeFile, tempFile, [, NULL, NULL, NULL, NULL, NULL, background[, class ]])

**Arguments**

printer	A string giving the printer name to print the document. If you want to print to a file rather than to a printer, set printer to NULL; the print file is placed in the directory specified by tempFile, or in the Applixware temporary directory if no directory is specified in tempFile.
colorFlag	Indicates whether the document is being printed on a color printer. If you are printing to a color printer, set colorFlag to TRUE; otherwise, set colorFlag to FALSE, which is the default.
copies	The number of copies to print. Default is 1.
bannerFlag	Indicates whether to include a banner page with the printed document. Set to TRUE if you want a banner page; FALSE if you don't, (which is the default).
allPagesFlag	Indicates whether to print all pages of the document. Set to TRUE to print all pages; FALSE if you want to print a range of pages.
startPage	A number indicating the first page to print if you are printing a range of pages. If allPagesFlag is TRUE, startPage is ignored.
endPage	A number indicating the last page to print if you are printing a range of pages. If allPagesFlag is TRUE, endPage is ignored.
mergeFile	The name of the file to merge with the document when printing. Specify NULL if you are not using a merge file.

tempFile	The name of the print file. This file is created to print the document and is deleted when printing is completed (unless printer is set to NULL). If you do not supply a path name for tempFile, the print file is created in the Applixware temporary directory.
background	Indicates whether to print the document ``in the background." Set to FALSE to not print in the background. If TRUE, a new axmain (an additional Applixware server process) will be created, without checking out a new license. That new process will be terminated upon completing the print job.
class	A constant indicating if printing will be performed on a PostScript or PCL5 device, as follows: PostScript PCL

**Description** Prints the current Macro Editor document using the attributes you specify. It is best to assign a FALSE value to background unless you are printing a very long document (approximately 20 pages or more). Otherwise, it may take longer to invoke the new Applixware process than to print in the foreground.

To collate copies (that is, cause multiple copies to print as sets rather than printing all page 1's followed by all page 2's, and so on), set the [Print Multiple Copies as Sets](#) (axPrintCollate) Printing preference to TRUE before calling ME\_PRINT@ as follows:  
CHANGE\_PREFS@("axPrintCollate", TRUE)

---

## ME\_REVERT@

---

Restores a Macro Editor document to its previous saved state

**Format** ME\_REVERT@()

**Description** Restores a Macro Editor document to its state when last saved. This eliminates all changes made to the document since the last save. The document remains open. Unlike File ® Revert, ME\_REVERT@ does not prompt to verify that the user really wants to restore the document, and discard any edits.

---

## ME\_SAVE@

---

Saves the current Macro Editor document

**Format** ME\_SAVE@()

**Description** ME\_SAVE@ writes the current Macro Editor document to a file. If the macro has never been saved before, it is saved to a file having the name displayed in the Macro Editor window title area.

The document is saved in the format in which it was read (which is either Formatted or ASCII format).

**See also** [ME\\_SAVE\\_AS@](#)

---

## ME\_SAVE\_COMPILED@

---

Saves the current Macro Editor document in compiled format

**Format** ME\_SAVE\_COMPILED@()

**Description** ME\_SAVE\_COMPILED@ writes the current Macro Editor document to a .elo file. If the macro has never been saved before, it is saved to the file name and path displayed in the Macro Editor window title area. A .elo extension is added to indicate the compiled format.

This macro is run when you click the Also Save Compiled format in the save as dialog of the Macro Editor.

**See also** [ME\\_SAVE\\_AS@](#)

---

## ME\_SAVE\_AS@

---

Saves the current file to a new name

**Format** ME\_SAVE\_AS@(name[, mode[, grpAccess[, allAccess] ] ])

**Arguments**

name	The current file's new name.
mode	If set to 0 or 1, the file is saved in compiled mode. Be careful. This can overwrite the original file. In most cases, you will want to execute the following code sequence: ME_SET_SAVE_COMPILED@(TRUE) ME_SAVE@()
grpAccess	A number indicating the read and write permissions for the file for a member of the same group. This can be one of the following values: 0 No read or write permissions for the file. (default) 1 Read permission for the file.

	2	Read and write permissions for the file.
allAccess	A number indicating the read and write permissions for any user. This can be one of the following values:	
	0	No read or write permissions for the file. (default)
	1	Read permission for the file.
	2	Read and write permissions for the file.

**Description** Renames the current file to name and saves it to disk. The file's access privileges are set using grpAccess and allAccess.

**See also** [ME\\_SAVE@](#)

---

## ME\_SELECT\_ALL@

---

Selects all text

**Format** ME\_SELECT\_ALL@()

**Description** Selects the entire document contents. WP\_SELECT\_ALL@ is called by Edit ® Select All.

---

## ME\_SET\_DEFAULT\_POINTER@

---

Sets the default pointer shape

**Format** ME\_SET\_DEFAULT\_POINTER@(pointerName)

**Arguments** pointerName The name of the new default pointer.

**Description** Selects the mouse pointer used by default. This is the pointer that is used within the Macro Editor's window excluding its margin. After using this macro, the default pointer is changed for all Macro Editor windows.

If you wish to always use this pointer, it can be made permanent by setting a preference.

To obtain a list of available shape names, use [GET\\_CURSOR\\_LIST@](#)

**See also** [ME\\_SET\\_TEXT\\_POINTER@](#)

---

## ME\_SET\_HOOK@

---

Executes a macro every time you open a Macro Editor document

**Format** ME\_SET\_HOOK@(macroName)

**Arguments** name           The name of the hook macro. If macroName is not loaded when this macro executes, ELF will load, compile, and execute a file with the same name (after a .am extension is appended) as macroName that exists in your default path.

**Description** Embeds within a Macro Editor document an instruction to run a given macro every time a Macro Editor document is opened. To unset the ME\_SET\_HOOK@ ``hook" in a Macro Editor document, run ME\_SET\_HOOK@ without any arguments.

---

## ME\_SET\_SAVE\_COMPILED@

---

Determine whether compiled object is also saved

**Format** ME\_SET\_SAVE\_COMPILED@(compiledFlag)

**Arguments** compiledFlag  
                  A Boolean value, where TRUE means save on next compile, FALSE means do not save on next compile.

**Description** Determines whether a Macro Editor document is saved on the next compile.

---

## ME\_SET\_SAVE\_FORMATTED@

---

Determine whether next save is formatted or ASCII

**Format** ME\_SET\_SAVE\_FORMATTED@(formattedFlag)

**Arguments** formattedFlag  
                  A Boolean value, where TRUE means save as a formatted document, FALSE means save as an ASCII document.

**Description** Determines whether a Macro Editor documented is saved as a formatted document or as an ASCII document.

---

## ME\_SET\_TEXT\_POINTER@

---

Sets the new text pointer shape

**Format** ME\_SET\_TEXT\_POINTER@(pointerName)

**Arguments** pointerName The name of the new text pointer.

**Description** Sets the new text pointer shape. This is the pointer that is used within the Macro Editor's margins. After using this macro, the text pointer is changed for all Macro Editor windows.

If you wish to always use this pointer, it can be made permanent by setting a preference.

To obtain a list of available shape names, use [GET CURSOR LIST@](#).

**See also** [ME SET DEFAULT POINTER@](#)

---

## ME\_TOGGLE\_DEBUGGER@

---

Turns debugger mode on and off

**Format** ME\_TOGGLE\_DEBUGGER@()

**Description** If debugger mode is off, ME\_TOGGLE\_DEBUGGER@ turns it on. If debugger mode is on, ME\_TOGGLE\_DEBUGGER@ turns it off. When debugger mode is on, the macro document cannot be edited and many of the menu options are grayed.

**See also** [ME IN DEBUG MODE@](#)

---

## ME\_UNDO@

---

Undoes the last edit in the current Macro Editor document

**Format** ME\_UNDO@()

**Description** Repeated use of ME\_UNDO@ will undo previous edits in the reverse order in which they were made. ME\_UNDO@ is called by Edit ® Undo.

For more information, see [Undo Description](#).

---

## ME\_VIEW\_EXPRESSLINE@

---

Toggles on the Macro Editor *Expressline*

**Format** ME\_VIEW\_EXPRESSLINE@()

**Description** Turns on and off the Applixware *Expressline* in the Macro Editor.

---

### \* --> Macro Editor

---

To edit or create macro documents with the Applixware Macro Editor, choose \* ® Macro Editor.

You can open and manipulate the macro documents generated when you create a macro just as you do other Applixware macro documents. In addition to many of the same editing options found in Applixware Words, the Macro Editor includes a verification and debugger facility, a dialog box editor, and a bitmap editor.

The menus available in the Macro Editor include:

\*

Contains general use, customizing, and the Macro Editor Preferences options.

File

Lets you manipulate the macro document as a file, for example, to save, print, delete, and send the document through Applixware Mail. Contains the options used to compile, save and install macros. Also lets you import and export files in ASCII format.

Edit

Contains copy editing functions such as cut and paste.

View

Lets you remove error messages from a compiled macro and control typeface, size, bold, and italic attributes.

Find

Used to search the macro document for text or errors and to move the cursor to specific places.

#### Format

Lets you enhance your macro document with headers, footers, margins and special characters.

#### Tools

Lets you run a debugger, install files explicitly and create and edit dialog box and bitmap images.

#### Help

Provides a display of information about the Applixware version you are currently using and a mechanism for getting help on menu options and key accelerators currently assigned.

The *ExpressLine* appears below the menu bar. The border around an *ExpressLine* button indicates that the operation is currently available. Some options require you to take certain actions, such as selecting an item, before the button is available. To use the *ExpressLine*, click on any active button. To learn what option each *ExpressLine* button represents, choose Help ® [On Context](#), then click on the button you want help with. You can insert icons for frequently used operations here.

See \* ® [Customize Menu Bar](#) for information on designing your own *ExpressLine*.

---

## Help --> On Macro Editor

---

Use the Macro Editor window to edit or display the commands that make up a macro document. The Macro Editor window works in the same way as the Words window.

The Macro Editor menus include:

\*

Contains general use, customizing, and the Macro Editor Preferences options.

#### File

Lets you manipulate the macro document as a file, for example, to save, print, delete, and send the document through Applixware Mail. Contains the options used to compile, save and install macros. Also lets you import and export files in ASCII format.

#### Edit

Contains copy editing functions such as cut and paste.

#### Find

Used to search the macro document for text or errors and to move the cursor to specific places.

## View

Lets you remove error messages from a compiled macro and control typeface, size, bold and italic attributes.

## Tools

Lets you run a debugger, install files explicitly, enhance your macro document with headers, footers, margins and special characters. Also lets you create and edit dialog box and bitmap images.

## Help

Provides a display of information about the Applixware version you are currently using and a mechanism for getting help on menu options and key accelerators currently assigned.

The *ExpressLine* appears below the menu bar. The border around an *ExpressLine* button indicates that the operation is currently available. Some options require you to take certain actions, such as selecting an item, before the button is available. To use the *ExpressLine*, click on any active button. To learn what option each *ExpressLine* button represents, choose Help ® **On Context**, then click on the button you want help with. You can insert icons for frequently used operations here.

See \* ® **Customize Menu Bar** for information on designing your own *ExpressLine*.

---

## Macro Editor --> Select Window

---

This message appears when you run a macro that requires a certain window but ELF cannot determine which window to use. Select the window you need the macro to use from the list of windows currently open on your desktop.

---

## View --> Delete Error Messages

---

To remove the error messages inserted into this macro document as a result of compiling, click Delete Error Messages. All message are deleted. You can also print the macro document to remove all the error messages.

See also **Compiler Error Messages**.

---

## Tools --> Debugger Mode

---



To preview and track the execution of a macro so you can determine potential and actual problem areas, choose Tools ® Debugger Mode.

This option appends a Debug window to the bottom portion to your Macro Editor window which you can use to diagnose the execution of your ELF statements.

Debugger Mode includes standard debugger features such as setting breakpoints, and single stepping macros.

If your macro calls sub-macros in other files, the debugger always displays the file containing the break or step point, unless you use multiple debugger windows. You may have up to 16 debugger windows, each focusing on a different macro document. The state of the macro is displayed in the status list box of each debugger window. Single stepping and continuing from breakpoints may be performed from any of the debugger windows although the setting of breakpoints and displaying of current statement is done only in the window containing the macro file where the statement is.

To turn off Debugger Mode, click on Tools ® Debugger Mode again. This automatically clears all breakpoints set in this debugger window and if this is the only debugger window, this also terminates the macro under debug.

#### Break

To set a breakpoint, select an executable line of source with the mouse and then press this button. Note that if you select a non-executing line such as a comment or VAR statement, the breakpoint will never be hit!

#### Clear

Clears the current breakpoint when in debugging mode.

#### Print

Displays the value of the variable that is selected in the macro document.

#### Cont

Begins or continues program execution.

#### ClrAll

Clears all breakpoints when in debugging mode.

#### Where

Pressing this button displays the current execution stack, telling you what macro called this macro and so on.

#### Step

While at a breakpoint, press this button to proceed to the next statement. The statement encountered will be displayed in the status list box and will be selected in this window if it is contained in this macro document.

#### Up

Pressing this button moves the debugger's display context up the execution stack and positions the debugger display pointer to the line that called the current routine. This is most often

used in conjunction with the Print button to display the caller's variables. It is also used to set additional breakpoints in the caller for when the current routine returns.

#### Run

Select the name of the macro to run and press this button. Alternately, you may invoke the macro by any other normal method such as selecting it from a menu bar or by the \* Ⓜ Run Macro. Regardless of how the macro is invoked, any breakpoints you have set will be stopped at.

#### Next

While at a breakpoint, press this button to proceed to the next statement within this macro. The Next operation does not step into sub-macros. The statement encountered will be displayed in the status list box and will be selected in this window if it is contained in this macro document.

#### Down

This is the inverse of the Up button. It has no meaning until the Up button has been pressed.

#### Quit

Terminates a debugging session.

#### Help

Displays help text for debugger activities.

Turn on Catch Errors if you want to jump to the specific code where the error was thrown.

---

### **Tools --> Install Macro File**

---

To explicitly install a macro document or file, choose Tools Ⓜ Install Macro File. Use this option when the macro file contains more than one macro or the macro in the document does not have the same name as the macro document.

You can install a macro document automatically when you log in using a login file.

See [login file](#) for more information.

---

### **Tools --> Edit Bitmap**

---

To edit a Macros bitmap image, click Edit Bitmap. Double-click the image you want to edit. Use File Ⓜ Exit to store the changes in the Bitmap Editor.

---

## login.am File

---

You can use an optional login.am file to automatically run macros when Applixware is started. This is useful for customizing Applixware start-up procedures to meet your specific needs. For example:

- Install macro documents that might not otherwise be installed at start-up.
- Display certain Applixware applications or documents when you start up Applixware.
- Include a custom start-up screen that displays a welcome document or informational message.

The login.am file contains a login macro that includes macros that are to be run at start-up. Upon start-up, Applixware checks your macros directory for a file named login.am. If the file exists, Applixware runs the LOGIN macro in the file as part of the start-up procedures.

See also [syslogin.am](#).

The following is a sample login.am file:

```
macro login
```

```
var default_dir, default_dir_size, i, str, which_axprof, print_value, axlang_value, i2, str2, retval
```

```
' set up the ELF search path to use default plus system_dir@
```

```
    default_dir = system_var@"elf_search_list@"
```

```
        ' gets default search paths
```

```
    default_dir_size = array_size@(default_dir)
```

```
        ' finds size of 0 based array
```

```
    default_dir[default_dir_size] = system_dir@()
```

```
        '-adds /install_dir/axdata
```

```
    set_system_var@"elf_search_list@",default_dir)
```

```
        'resets it
```

```
        for i = 0 to default_dir_size
```

```
            str = str++"\n"++default_dir[i]
```

```
        next i
```

```
' get info from ax_prof4 and display with elf_search_list@
```

```
    print_value = preferences@"axPrintShellCmd")
```

```
    axlang_value = preferences@"axLanguage")
```

```
        ' shows which dictionary to use
```

```
info_message@("login.am executed\naxLanguage is: " ++ axlang_value ++ "\naxPrint-ShellCmd is: " ++ print_value ++ "\n" ++ str2 ++ "\nelf_search_list is:" ++ str)
```

```
endmacro
```

---

## syslogin.am File

---

You can create a optional syslogin.am macro. The syslogin.am macro file should be placed in the /install\_dir/axlocal/eng directory, where /install\_dir is the directory where you installed Applixware. Regardless of where the syslogin.am macro file resides it will be executed before the [login.am](#) macro file.

See also [Search Paths](#).

If the syslogin.am macro is placed in the /install\_dir/axlocal/eng directory it will affect all Applixware users. If it is placed in /user/axhome/macros it will only affect the individual user.

To create a syslogin.am file:

1. Start the Macro Editor by choosing the \* ® Macro Editor.
2. Type the following statement as the first line of the syslogin.am macro:

```
macro syslogin
```

3. Following the macro statement, include the macro calls that you want to run at start-up.
4. Type the following statement as the last line in document:

```
endmacro
```

5. Choose File ® Save As.
6. Save the file as syslogin, in your /install\_dir/axlocal/eng directory.

After you install the macro and edit your syslogin.am file, have all users exit from Applixware. When users restart Applixware, the macro executes automatically.

---

## Search Paths

---

The sequence of directories that Applixware looks in for a command or file is called the *search path*. The following is the default search path for ELF:

1. The /user\_dir/axhome/macros or equivalent user-specific directory
2. The /install\_dir/axlocal directory
3. The /install\_dir/axlocal/eng directory

4. The /install\_dir/axdata/eng/Demos directory
5. The /install\_dir/axdata/eng directory
6. The /install\_dir/axdata/elf directory

If you are using an interface language other than English, then the eng portion of these pathnames will be replaced by the corresponding language abbreviation.

You can customize your search path by setting a [UNIX system variable](#).

The elf\_search\_list@ system variable contains a list of pathnames that replace the default ELF search path. When set, ELF looks at the directories named in this variable for macros, image files, dialog boxes, object files, and so on.

For example:

```
names = "/user/john/macros", "/ax/lpgms", "/ax/pgms/axelf"  
set_system_var@("elf_search_list@", names)
```

This statement says that ELF should look in the three named directories for information and in only these three named directories.

See also [SET SYSTEM VAR@](#).

---

## View --> Zoom

---

You can display the document onscreen in a variety of magnification sizes, larger or smaller. In addition, you can decide the degree of accuracy you need in your display, with the possible tradeoff of a slight decrease in speed for more accuracy.

To display the text in a different size, choose one of the magnification size options under View ® Zoom:

- 40%
- 60%
- 80%
- 100%
- 120%
- 150%
- 200%
- 300%
- 400%

Sizes less than 100% will shrink the display of the text; sizes greater than 100% magnify the text. 100% displays the document at full size.

The Zoom size does not affect the printed document. The document always prints at 100% size.

---

---

---

## **Format --> Character Settings**

---

To set a local typing attribute or to change the attributes of the selected text, choose Format ® Character Settings.

If no text is selected, text you type is displayed in the settings you choose here. These settings remain in effect until you move the text cursor using the cursor keys or clicking with the mouse.

### Typeface

Choose the typeface to use for the text.

### Size

Choose the type size (in points) to use for the text.

To specify a non-standard size, type the point size you want.

### Color

Choose the color to use for the text.

### Position

Choose the vertical position to place the text on, above, or below the baseline, or normal text line:

#### Baseline

Positions characters on the normal line of text.

#### Subscript

Displays text below the baseline. In the by \_\_\_ points entry area, click and type the number of points to lower the text from the baseline.

#### Superscript

Displays text elevated from the baseline. In the by \_\_\_\_ points entry area, click and type the number of points to raise the text from the baseline.

#### Underline

Choose the underlining option to use for the text:

None - Turns off underlining.

Single - Turns on single underlining.

Double - Turns on double underlining.

Single Words - Single underline each word, not spaces between.

Double Words - Double underline each word, not spaces between.

Single Not Tabs - Single underline for each word, not tabs between.

Double Not Tabs - Double underline for each word, not tabs between.

#### Bold

Turn on to display text in a bolded typeface.

#### Italics

Turn on to italicize text.

#### Strike Through

Turn on to bisect text with a horizontal line.

#### Hyphenation Allowed

Turn on to allow hyphenation within the text.

The All As Is setting for a menu option or a greyed toggle indicates that the selected text includes different settings for that attribute. Applying the dialog box has no effect on the mixed settings. You can select a specific setting to override the mixed attributes.

To change default text attributes for the entire document, use the Normal style.

---

## Format --> Paragraph Settings

---

To display or change the attributes used by paragraphs, choose Format ® Paragraph Settings.

You can also set paragraph indents using the graphical ruler shown with View ® **Ruler**, and justification using the **Justification** buttons on the *ExpressLine*.

Any option that is greyed indicates that the selected paragraphs have mixed settings for this attribute. You can type a setting in the greyed entry box to set the attribute the same for all selected paragraphs.

Select any of the following topics for more information:

[Hyphenation](#)

[Indents](#)

[Justification](#)

[Levels](#)

[Pagination and Spelling Options](#)

[Paragraph and Line Spacing](#)

---

### **Format --> Promote**

---

To decrease the level number of the current paragraph by one level, choose Format ® Promote.

To set paragraph levels, see Format ® [Paragraph Settings](#).

If Level Affects Indent is turned on in the Paragraph Settings, the level of the paragraph determines the number of tab stops to indent the paragraph.

---

### **Format --> Demote**

---

To lower the current paragraph by one level, choose Format ® Demote.

To set paragraph levels, see Format ® [Paragraph Settings](#). The level of the paragraph can affect the indents.

---

### **File --> Page Setup**

---

To define the page layout for the document, choose File ® Page Setup.

Set the following features and dimensions:

Paper Size

Sets the Width and Height of the paper you are using to print, expressed in the current units (inches or centimeters).

Orientation

Turns the page vertically or horizontally in the on-screen display and in the printed document.

To print the page with the Height as the vertical dimension, click on Portrait.

To print the page with the Height as the horizontal dimension, click on Landscape.

### Facing Pages

Creates different page margins for odd and even pages.

The margins on the sides of the page change with the Facing Pages setting: when Facing Pages is turned off, the side margins are labeled Left and Right. When Facing Pages is turned on, the side margins are labeled Inner and Outer.

### Margins

The Margin settings define the amount of space between the text and the edges of the paper, expressed in the current units.

### Apply

Choose whether you want these settings to apply to the Whole Document, from This Point Forward, or the Current Section.

---

## **File --> Header & Footer**

---

To place text at the top and bottom of every page, choose File ® Header and Footer. You can create different headers and footers for first, last, odd and even pages.

### Header

Click on Header to place a header at the top of every page. If you do not choose any other options, the header is the same throughout the document.

Click on First Page Different to create a header for the first page.

Click on Last Page Different to create a header for the last page.

Click on Odd and Even Pages Different to create alternating headers for even and odd pages.

### Distance from Page Top

Enter the distance from the top of the page in the current units at which you want the header to appear. This setting is document-wide.

### Footer

Click on Footer to place a footer at the bottom of every page. If you do not choose any other options, the footer is the same throughout the document.

Click on First Page Different to create a footer for the first page.

Click on Last Page Different to create a footer for the last page.

Click on Odd and Even Pages Different to create alternating footers for even and odd pages.

### Distance from Page Bottom

Enter the distance from the bottom of the page in the current units at which you want the footer to appear. This setting is document-wide.

The scrolling list box contains all the header and footer options you have created. To delete a header or footer option, click on the corresponding button to turn off that option.

To change the links between the header or footer and its counterpart in the previous section, select the header or footer option and click on the Link or Unlink button:

#### Link to Previous Section

Click here to link the currently selected header or footer to the previous section.

This option becomes available if you have preceding sections in the document, and the header or footer selected is not linked.

#### Unlink from Previous Section

Click here to unlink the currently selected header or footer from the previous section.

This option becomes available if you have preceding sections in the document, and the header or footer selected is linked.

---

### **Edit --> Shift Case**

---

To change the case of selected text, choose Edit ® Shift Case. Choose this option again if needed to change to the case you want.

The first word in the selected text determines the initial case setting. The case changes in the following order:

UPPER CASE

Initial Capitals

lower case

That is, if you select UPPER CASE text and then choose Edit ® Shift Case, the text changes to Initial Capitals. Choose Edit ® Shift Case again to change the text to lower case. The next time you choose Edit ® Shift Case, the text cycles back to UPPER CASE, and so on.

If the selected text is lower case to begin with, the first change is to UPPER CASE.

Mixed case, such as Applixware, changes to lower case (Applixware). The original mixed case cannot be retrieved using Edit ® Shift Case. Choose Edit ® Undo to return to mixed case.

---

## Edit --> Select All

---

To select all material in the document, choose Edit ® Select All. Selected material is highlighted, and includes the entire main flow but excludes headers and footers. Remember that even material not on the screen is selected.

---

## Compiler Error Messages

---

The ELF compiler in the Macro Editor sometimes throws "Macro too big" error messages. The following lists some of these messages and why they may occur:

<b>Error Message</b>	<b>Cause</b>
Macro too big, too many parse nodes	This is a serious compiler error and should rarely occur. Contact Applixware Customer Support at 1-800-8Applixware.
Macro too big, too many local variables	There is a limit of approximately 900 local variables in a single macro.
Macro too big, symbols frame overflow	Your program is too "rich" and is calling too many things or contains too many small functions. Or your program may contain too many literal strings or arrays.
Macro too big, code area overflow	Too much executable code. Convert duplicate code to subroutine calls. Use block assignments rather than explicit array element assignments. Use intermediate variables rather than complex array references.
Macro too big, too many symbols	Your program is too "rich" and is calling too many things or contains too many small functions.

---

---

---

## File --> Save As

---

Click Save As to save this macro document under a different name. You can also use Save As to change the file format or to name an unnamed macro document. Save As automatically appears the first time you try to save a macro document so you can name it.

Saving a macro document does not automatically compile, verify the syntax, or install it. Use File ® **Compile & Save**, to perform all these operations in one step. Or, use File ® **Compile** to check your macro syntax before you save and install it.

The list box shows only the macro documents and directories currently stored in this location.

### File

The macro document's current name and directory path are displayed here. Click here to edit it or enter a new one.

Generally macro names are the same as the macro document in which they reside. If the macro name is different from its macro document name, you must explicitly install the macro using Tools ® **Install Macro File** so Applixware can find it.

Be sure the macro document name is unique or the existing macro document will be overwritten.

Macro names must follow these conventions:

- Up to 30 alphanumeric characters
- The first character must be a letter.
- Macro names cannot end with @ or \$ as they are reserved for by Applixware.
- Macro names are case insensitive.
- No spaces are allowed. Underscores are allowed.
- Certain words are reserved for ELF usage. If you use an ELF reserved word, a message appears so you can change it.

### Show Directories

If the Show Directories button is on, directory folders appear in the directory display. If the Show Directories button is off, files appear in the directory display, but directories do not.

### Save Formatted

Press this button to save the macro file in Applixware Words format. Turn this button off the save the macro file in ASCII format.

### Also Save Compiled

You can save ASCII or Formatted Macro documents in a Compiled format by turning on the Also Save Compiled option on the File ® Save As dialog box. Macro documents stored in compiled format do not have to be compiled before they are used.

Compiled files are machine-independent. You can use a macro stored in Compiled format on a different machine. Compiled files cannot be opened or edited by the macro editor.

Macros stored in compiled format have a .elo extension.

When both the Also Save Compiled and Save Formatted options are on, the macro is saved in both formatted and compiled format.

### Group

To assign read or read/write permission only to users in your operating system group, click Read and/or Write buttons on the Group line. Click again to remove the appropriate permission.

### Others

To assign read or write permission to any user, click Read and/or Write buttons on the Others line. Click again to remove the appropriate permission.

If neither lines have anything chosen, you are the only one who can access the macro document.

---

## File --> Compile & Save

---



To compile and save a macro document one step, click on File ® Compile and Save. Macros are automatically stored in the macros directory specified in Q ® Macro Editor Preferences **Macros Files Directory** option.

If a macro has a name different from the macro document where it resides, you must explicitly install it using Tools ® **Install Macro File**. You can also save or compile your macros separately with File ® **Save** or File ® **Compile**.

If a compilation error occurs, a message displays with the number of errors found. The errors are inserted into the file in a different typeface beneath the line of text where the error occurred. Use Find ® **Next Error** or Find ® **Previous Error** to quickly move to errors. You can remove all error messages in the macro document with View ® **Delete Error Messages**. Or, print the macro document to remove the error messages.

If errors are found, the macro document is saved but not installed. Fix the errors then re-compile before you install it either manually or with this option.

See also: **Compiler Error Messages**

---

## File --> Compile

---

To verify your macro syntax before you install it, choose File ® Compile.

If a compilation error occurs, a message displays with the number of errors found. The errors are inserted into the file in a different typeface beneath the line of text where the error occurred. Use Find ® [Next Error](#) or Find ® [Previous Error](#) to quickly move to errors. You can remove all error messages in the macro document with View ® [Delete Error Messages](#). Or, print the macro document to remove the error messages.

Or, use File ® [Compile & Save](#) to do all operations in one step.

See also: [Compiler Error Messages](#)

---

## File --> Send

---



To send the current document through Applixware Mail with a mail message, choose File ® Send, or click on the Send icon on the *ExpressLine*.

You can send a document in any format accepted by other software for which you have Applixware filters installed on your system. You can choose to localize linked Objects or send the insets as separate documents as attachments. The menus available in the Send Document dialog box include:

### Edit

Lets you Cut, Copy, and Paste text within the message you are composing.

### View

Turns on or off display of the *ExpressLine*.

### Lists

Lets you create a personal user list, personal and global aliases and distribution lists, display the contents of a distribution list or alias, and define recipient preferences. To define recipient preferences, you must have access to system files.

### Attachments

Lets you attach files and audio messages. When you attach files, the Attachments button in the *ExpressLine* (the paper clip icon) changes to reverse video.

### Tools

Allows you to spell check messages before sending, specify an alternate reply recipient (to allow the recipient to reply to someone other than you), attach your signature to the message, or jump to somewhere to do something. You can also save and restore messages.

Send the document and create the mail message using the following options:

#### To

Type the user name(s) of the recipient(s), or double-click on the user name(s) in the global or personal list box to add to Recipient list box.

#### Cc

Click Cc and type the username(s) of anyone to whom you want to send a copy of the document and message.

#### Bc

To send a blind copy of the document to the recipient, click Bc. Only the recipient sees that they have received the document.

#### Recipient list box

Displays the names you type or select to receive the document, preceded by To, Cc, or Bc to show the type of message each recipient will receive.

#### User list box

Displays the **personal** or **global** list of users for you to select recipients of the document. Double-click on the name of the person or list to whom you want to send the document.

#### Name

Type the name of the person or list to whom you want to send the document. Click Add to add the name to the recipients list.

#### Subject

The name of the document you are sending is the default subject of your mail message. To change the subject, click in the Subject entry area and edit the text.

#### Message

Type the text of the mail message. You can cut, copy, and spell check text within the message, and paste text from other sources.

#### Format

Select the format in which to send the current document from the pull down menu. The options you see are determined by the Applixware Filter Packs you have on your system for the current application. If you send the document in a format other than Applixware, the document you send is converted to the format required by the software you select.

To send the document in the format defined by the Recipient Preferences for each user, select From Profile (the default). If no Recipient Preferences are defined, the default is to send the document in Applixware format.

To override the Recipient Preferences, select a specific format.

#### Localize All Linked Objects

This option becomes available when you are sending an Applixware Words document that contains linked Graphics or Spreadsheets insets. You can choose whether to localize the insets or send the insets as separate Graphics or Spreadsheets documents. By default, Localize All Linked Objects is on.

To send linked insets as separate, attached documents, click on Localize All Linked Objects to turn the option off.

#### Reply by

Adds a line requesting a reply at the top of the mail message. The Reply by entry area becomes available for you to enter text.

#### Urgent

Sends the document with an urgent icon, an exclamation point (!) that appears in the recipient's Inbox.

#### Certified

Sends you a return message to notify you that the recipient has read your mail message. A pencil status icon appears next to your copy of the message in the Outbox to show that you sent the message as certified.

#### Encrypt Message

Attaches a password to the message. The recipient cannot read the message without typing the password. Passwords are case sensitive.

#### Copy to Outbox

Saves a copy of the message to your Outbox. You can save a copy of all your messages by turning on this option in your Q ® Mail Preferences.

#### Add Recipients to Personal List

Adds the recipients of this message to your personal user list. Applixware checks for duplicates before adding the name to your list. You can automatically add the names of all recipients of your messages by turning on the [Add Recipients to Personal List](#) option in Q ® [Mail Preferences](#).

---

## Personal

---

To display your personal user list in the user list box, click personal. If you create a personal user list, it is displayed by default. Your personal user list also includes your Personal Alias List.

---

## Global

---

To display the global user list in the user list box, click global. If you have not created a personal user list, the global user list is displayed.

---

## Edit --> Undo

---

To cancel the action you just took, click Edit ® Undo.

You can walk back through your actions up to the last save using Undo repeatedly. Edit ® **Redo** restores any action you ``undid."

To activate this option you must first take an action in the macro document.

---

## Edit --> Redo

---

To restore an action you have just cancelled with Undo, click Redo. Redo can also walk back through your ``undone" actions.

To activate this option you must first Edit ® **Undo** an action.

---

## Edit --> Cut

---



To remove selected material from the macro document and move it to the clipboard, click Cut. Material remains on the clipboard until you overwrite it by cutting or copying other material. Use Edit ® **Paste** to put material from the clipboard or operating system window into the macro document.

To activate this option, you must first select some text.

---

## Edit --> Copy

---



To put a copy of selected material onto the clipboard, click Copy. Material remains on the clipboard until you overwrite it by copying or cutting other material. Use Edit ® **Paste** to put material from the clipboard into a macro document. You can also paste using your window manager paste function into an Applixware document.

To activate this option, you must first select some text.

---

### Edit --> Paste

---



To paste material on the clipboard into your macro document, click Paste. Material is pasted at your text cursor. You can repeatedly paste the same material, since the clipboard remains intact until more material is cut or copied onto it.

To activate this option, you must first have something cut or copied, or have an external file ready to paste.

---

### Edit --> Delete

---



To delete selected material, click Delete. Deleted material is removed from the macro document. It is not pasted on the clipboard. If you delete material accidentally, click Edit ® **Undo** to retrieve it. Once you use Save, or Save As, you can no longer retrieve deleted material.

To activate this option, you must first select something to delete.

---

### Find --> Next

---

To advance to the next occurrence of search text specified in Find ® Next. Click Find ® Previous to move back.

---

### Find --> Previous

---

To retreat to the previous occurrence of search text specified in Find ® Previous. Click Find ® Next to advance through the macro document.

---

### Find --> Page

---

To move to a specific page in a paginated document, choose Find ® Page Number.

Page Number

Type the number of the page where you want to move.

---

## Find --> Line

---



To move the text cursor to a certain line number in your document, choose Find ® Line and enter a number between 1 and 99999. The cursor moves to the beginning of that line.

---

## Find --> Next Error

---



To move forward to the next error in the macro document that was generated as a result of compiling, click Next Error. You can compile using File ® [Compile](#) or File ® [Compile & Save](#).

This option does not wrap around through the start of the macro document. Use Find ® [Previous Error](#) to move back to an error message located between your cursor position and the beginning of the macro document.

Use View ® [Delete Error Messages](#) to remove all of them from the document.

---

## Find --> Previous Error

---



To move back to the previous error in the macro document that was generated as a result of compiling, click Previous Error. You can compile using File ® [Compile](#) or File ® [Compile & Save](#).

This option does not wrap around through the end of the macro document. Use Find ® [Next Error](#) to forward to an error message located between your cursor position and the end of the macro document.

Use View ® [Delete Error Messages](#) to remove all of them from the document or print it.

---

## View --> Boundaries

---

To display all formatting boundaries in the document, choose View ® Boundaries.

The following boundaries may display if they exist in your document:

Margins

Headers and footers

Sections

Columns  
Table cells

---

---

---

## Macro Editor Preferences

---

Double-click on any of the following topics for more information.

[Macros Files Directory](#)

[Cursor Color](#)

[Template](#)

[Save Aster\\*x 2.1 and New Macros as](#)

[Macro to Run when Starting the Macro Editor](#)

[Pointers: Macro Editor](#)

[Pointers: Recordable Functions](#)

[Pointers: Unrecordable Functions](#)

Choose [Applixware Preferences](#) to see help on all categories of preference options.

---

### Save Aster\*x 2.1 and New Macros as (meSaveAscii)

---

This option sets the default format to save a new Applixware or Aster\*x 2.1 Macro Editor document. If you have existing Aster\*x 2.1 macros, or are creating new macros you can specify the file format type for all new macros or recently edited macros.

Existing Applixware documents have a Save format attached. The Save format can be changed for an individual macro document.

Choose Formatted .am Files to save your macro document in Words (.aw) format. This format allows you to open your macro document in Words, use multiple fonts, include insets, and use Word macros to manipulate data. For example, you could create a list of hyperlinks to your macros and place these hyperlinks at the beginning of your document. Clicking on the hyperlink would jump you to your macro.

Choose ASCII .am Files to save your macro documents in ASCII format.

The default is Formatted .am Files.

---

### **Pointers: Recordable Functions (axElfRecordPointer)**

---

Use this option to specify the type of pointer used when you are making a keystroke recording.

By definition, the pointer type specified here should be different from [Pointers: Unrecordable Functions](#) and from your default cursor type. Use the scrolling list to select a cursor type.

The default cursor is a dot inside a box.

---

### **Pointers: UnRecordable Functions (axElfNoRecordPointer)**

---

Use this option to specify the type of pointer used when you are trying to make a keystroke recording but you have asked for something that cannot be recorded.

For example, you cannot use the mouse to select objects when recording so if you try to record that action, the mouse pointer changes to the shape specified here to remind you that no recording is taking place.

By definition, the pointer type specified here should be different from the [Pointers: Recordable Functions](#) and from your default cursor type. Use the scrolling list to select a cursor type.

The default cursor is a dot.

---

### **Macro Files Directory (axMacrosDir)**

---

This is the default directory where your ELF macros are stored. To designate the directory where Applixware looks for macros, enter it in the Macro file directory.

Enter the directory path name in the entry area.

The default directory is `//login_dir/axhome/macros`, where `login_dir` is the name of the directory where Applixware is installed.

---

### **Cursor Color (meCursorRGB)**

---

Use this option to control the cursor color in a Macro editor document. This is useful if you have difficulty spotting your cursor on the screen. The RGB value is specified as a 255-based, comma-separated number. For example, to display a red cursor, enter the value 255,0,0.

You need to log out of Applixware before this setting will take effect.

---

### **Macro to Run when Starting the Macro Editor (meElfHook)**

---

This option sets a macro to run automatically when you open a Macro Editor window or document. Enter the macro name in the entry area.

---

### **Template (meNewTemplate)**

---

This option shows the current template used to create a new Macro Editor document. By default, the template most recently selected is used for the new document. If no template has been selected it uses the `empty.am` template.

To specify a different new template, enter the complete path of the template document.

---

### **Pointers: Macro Editor (meTextPointer)**

---

To change the pointer type used in the Macro Editor or the directory where your macros are stored select the pointer type you want to use from the display box. Use the scroll bar to see more pointer types.