Applixware Mail
ELF Reference

This manual was produced using Applixware.

Printed: June 2010

MAIL_ADD_ATTACHMENT@

Attaches a file to a message

Format flag = MAIL_ADD_ATTACHMENT@(fileName)

Arguments fileName   The full path name of the file to attach.

Description Attaches a file to the current message. The macro checks for the existence of the file. If the file does not exist, it returns FALSE. If the file exists, it attaches the file and returns TRUE.

MAIL_ADD_BC_RECIPIENT@

Adds a blind copy recipient to a message

Format flag = MAIL_ADD_BC_RECIPIENT@(userName)

Arguments userName   The user name to add as a blind copy recipient of the message.

Description Adds a user as a blind copy recipient to the recipient list of the message being constructed. If userName is added successfully, it returns TRUE; otherwise it returns FALSE.

See also **MAIL_ADD_CC_RECIPIENT@**
MAIL_ADD_TO_RECIPIENT@

MAIL_ADD_CC_RECIPIENT@

Adds a carbon copy recipient to a message

Format flag = MAIL_ADD_CC_RECIPIENT@(userName)

Arguments userName   The user name to add as a carbon copy recipient to the message.

Description Adds a user as a carbon copy recipient to the recipient list of the message currently under construction. If userName is added successfully, it returns TRUE, otherwise it returns FALSE.

See also **MAIL_ADD_BC_RECIPIENT@**

MAIL_ADD_TO_RECIPIENT@

MAIL_ADD_ENV_FIELDS@

Adds custom lines to the message envelope currently being created

Format flag = MAIL_ADD_ENV_FIELDS@(lineArray)

Arguments lineArray   The text being added to the message envelope. This argument must be an array.

Description Adds custom lines to the message envelope currently being created. Mail assumes that this information is correctly formed and does not perform any edit checking on this data.

FALSE is returned if lineArray is not an array. Otherwise, this macro returns TRUE.

MAIL_ADD_MESSAGE@

Appends information to the current message

Format flag = MAIL_ADD_MESSAGE@(msgArray, where)

Arguments msgArray   An array of strings to append to the current message body.

where   If where = 0, the msgArray is added to the beginning of the current message.

If where = 1, the msgArray is added to current message at the cursor position.

if where > 1, the msgArray is appended to the end of the current message.

Description Appends information in the msgArray to the current message body. If the information is appended successfully, it returns TRUE; otherwise it returns FALSE.

This macro works only with new, forwarded, or reply mail messages.

MAIL_ADD_SUBJECT@

Adds subject to Send Mail dialog box

Format flag = MAIL_ADD_SUBJECT@(subject)

Arguments subject   The subject of the message being created.

Description Adds subject to the message being created. If subject is added successfully, it returns TRUE; otherwise it returns FALSE.

MAIL_ADD_TO_FOLDER@

Adds message(s) to the top of a folder

Format flag = MAIL_ADD_TO_FOLDER@(format **inb_item** message, destination, createFlag)

Arguments message   The message(s).

destination   The destination folder.

createFlag   If the folder doesn't exist and createFlag is set to TRUE, the folder is created. If createFlag is FALSE, an error is thrown.

Description Adds the message(s) to the top of the target folder. If message was successfully added to destination, TRUE is returned. If message was not added to destination, FALSE is returned.

MAIL_ADD_TO_RECIPIENT@

Adds original recipient names to Send Mail dialog box

Format flag = MAIL_ADD_TO_RECIPIENT@(userName)

Arguments userName   The user names to add as recipients to the Send Mail dialog box.

Description Adds userName to the recipient list in the Send Mail dialog box. If userName is added successfully, it returns TRUE, otherwise it returns FALSE.

See also **MAIL_ADD_BC_RECIPIENT@**

MAIL_ADD_CC_RECIPIENT@

MAIL_ADDRESSPARSE@

Takes an address string and returns it as an array of addresses

Format returnAddressArray = MAIL_ADDRESSPARSE@ (addressesString)

Arguments addressesString

A string containing a list of addresses; addresses are separated from one another by commas.

MAIL_ATTACHED_FILES@

Returns a list of attached files

Format fileList = MAIL_ATTACHED_FILES@(format **inb_item** message[, folder])

Arguments message   A message structure.

folder   The folder containing the message. If this parameter is omitted, assume that the message is in the current folder.

Description Returns a list of attached files for message.

See also **MAIL_GET_SELECTED_ATTACHMENT@**
MAIL_HAS_ATTACHMENTS@
MAIL_SAVE_ATTACHMENT@

MAIL_CHANGE_FOLDER@

Makes folder current

Format flag = MAIL_CHANGE_FOLDER@(folder)

Arguments folder   A folder.

Description Makes folder the current folder. If successfully changed to folder, it sets the folder type ("private" or "shared"), and returns TRUE, otherwise it returns FALSE.

See also **MAIL_LIST_OF_FOLDERS@**

MAIL_COLOR_MESSAGE@

Colors an Inbox message

Format MAIL_COLOR_MESSAGE@(format **inb_item** message, sourceFolder, color)

Arguments message   The message.

sourceFolder The source folder.

color   The message color. The color values are passed as strings. Values you can use are:

red	Red
dkred	Dark Red
green	Green
dkgreen	Dark Green
blue	Blue
dkblue	Dark Blue
cyan	Cyan
dkcyan	Dark Cyan
magenta	Magenta
dkmagenta	Dark Magenta
yellow	Yellow
dkyellow	Dark Yellow
white	White
dkgray	Dark Gray
ltgray	Light Gray
X	Default color

Description Colors an Inbox message. The macro makes sure the message, source folder, and color exist before coloring the message.

See also **MAIL_COLOR_SELECTED_MESSAGES@**

MAIL_COLOR_SELECTED_MESSAGES@

Colors selected messages

Format MAIL_COLOR_SELECTED_MESSAGES@(color)

Arguments color   The message color. The color values are passed as strings. Values you can use are:

red	Red
dkred	Dark Red
green	Green
dkgreen	Dark Green
blue	Blue
dkblue	Dark Blue
cyan	Cyan
dkcyan	Dark Cyan
magenta	Magenta

dkmagenta	Dark Magenta
yellow	Yellow
dkyellow	Dark Yellow
white	White
dkgray	Dark Gray
ltgray	Light Gray
X	Default color

Description Colors the selected messages.

See also **MAIL_COLOR_MESSAGE@**

MAIL_COMBINE_QUERY@

Selects mail messages based on the
 specified criteria

Format MAIL_COMBINE_QUERY@(lnames, urgentFlag, unreadFlag, taggedFlag, attachFlag, documentFlag, audioFlag, kType, keyword, matchCaseFlag, dType, date1, date2)

Arguments lnames   A sender or recipient. This value must be an array even if it contains only one name.

urgentFlag   Boolean value where TRUE means find urgent messages.

unreadFlag   Boolean value where TRUE means find unread messages.

taggedFlag   Boolean value where TRUE means find tagged messages.

attachFlag   Boolean value where TRUE means find messages with attached files.

documentFlag
                Boolean value where TRUE means find messages that are mailed Applixware documents.

audioFlag   Boolean value where TRUE means find messages that contain audio components.

kType   One of the following type of keyword queries. The header file containing the DEFINE statements for the kType values is mail_.am.

MAIL#SUBJECT_	Keyword in subject
MAIL#SENDER_	Keyword in sender
MAIL#DATE_	Keyword in date
MAIL#COMMENT_	Keyword in comment
MAIL#BODY_	Keyword in message body

keyword   The keyword for the query.

matchCaseFlag
Boolean value where TRUE means query by case of keyword, otherwise the query is case insensitive.

dType
One of the following types of date queries. The header file containing the DEFINE statements for the type values is mail_.am.

MAIL#ALL_DATES_   All dates
MAIL#YESTERDAY_   Yesterday
MAIL#TODAY_   Today
MAIL#DATE_RANGE_   Range of dates, from date1 to date2

date1
The start date for a date range query. The date is in one of the **DATE_FORMAT@** formats. The formats can optionally include a time after the date.

date2
The end date for a date range query. The date is in one of the DATE_FORMAT@ formats. The formats can optionally include a time after the date.

Description Queries the Inbox based on the specified criteria. This criteria includes sender, recipients, keywords, dates, and status. After the query executes, only the messages fulfilling your criteria are displayed. If you wish all messages to again be displayed, use **MAIL_DISPLAY_ALL@**.

See also **MAIL_DATE_QUERY@**

MAIL_KEYWORD_QUERY@

MAIL_SENDER_QUERY@

MAIL_STATUS_QUERY@

MAIL_COMMENT_MESSAGE@

Adds or changes a message's comments

Format MAIL_COMMENT_MESSAGE@(format **inb_item** message, folder, comment)

Arguments message   The message.

folder   The folder.

comment   The comment.

Description Adds or changes a comment in the passed message. The comment appears at the top of the envelope in the message area.

MAIL_COMMENT_SELECTED_MESSAGE@

Adds comments to the current message

Format   MAIL_COMMENT_SELECTED_MESSAGE@(comment)

Arguments   comment   The comment text to be applied to the selected messages.

Description   Adds or changes the comments to the currently selected messages. This macro displays the Edit Comment dialog box, and is called by the Edit ® Comment menu option. The comment appears at the top of the envelope in the message area.

This is a User Interface macro.

MAIL_COPY_MESSAGE@

Copies a message to a folder

Format   MAIL_COPY_MESSAGE@(format **inb_item** message, source, destination)

Arguments   message   The message being copied.

source   The folder containing the message.

destination   A private or public destination folder.

Description   Copies messages from one folder, the source, to another, the destination. destination can be a private or shared folder. This macro will not let you:
·   Place more than one copy of a message in the same folder.

·   Copy a message to or from the Outbox folder.

See also   **MAIL_COPY_SELECTED_MESSAGES@**

MAIL_COPY_SELECTED_MESSAGES@

Copies selected messages to a folder

Format   MAIL_COPY_SELECTED_MESSAGES@(target)

Arguments   target   The target folder.

Description Copies selected messages from the current folder to the target folder.

This macro is called by the File ® Copy to Folder menu option.

See also **MAIL_COPY_MESSAGE@**

MAIL_CREATE_FOLDER@

Creates a new mail folder

Format flag = MAIL_CREATE_FOLDER@(folder)

Arguments folder The new folder name.

Description Creates a new mail folder whose name will be folder. If folder exists or if folder cannot be created, this macro returns FALSE. Otherwise, folder is created, added to the user's Inbox, and the macro returns TRUE.

This macro is called by the Folder ® Create Folder menu option.

MAIL_DATE_QUERY@

Queries Inbox messages by date

Format MAIL_DATE_QUERY@(type, date1, date2)

Arguments type One of the following types of date queries. The header file containing the DEFINE statements for the type values is mail_.am.

MAIL#ALL_DATES_   All dates
MAIL#YESTERDAY_   Yesterday
MAIL#TODAY_   Today
MAIL#DATE_RANGE_   Range of dates, from date1 to date2

date1 The start date for a date range query. The date is in one of the **DATE_FORMAT@** formats. The formats can optionally include a time after the date.

date2 The end date for a date range query. The date is in one of the DATE_FORMAT@ formats. The formats can optionally include a time after the date.

Description Queries the Inbox messages by date. The query can be for all dates, yesterday, today, or a range of dates. The messages in the current folder that meet the date query are displayed.

A query acts on the messages that are displayed in the current folder. A second query acts on the results of a first query, not on all the messages in the folder.

The macro is called by the Find ® By Date option.

See also   **MAIL_COMBINE_QUERY@**

MAIL_DELETE_ATTACHMENT@

Deletes a file attached to a message

Format   successFlag = MAIL_DELETE_ATTACHMENT@(folder, format **inb_item** message, filename)

Arguments   folder   The name of the folder in which the message is stored.

message   The name of the message containing the attachment.

filename   The file name of the attachment.

Description   Deletes a file attached to a mail message. This message is not only deleted from the message, it is deleted from the file system.

To obtain a list of the files attached to a mail message, use the **LIST_OF_FILES@** macro pointed at the file directory named in the message's format.

MAIL_DELETE_FOLDER@

Deletes a folder

Format   flag = MAIL_DELETE_FOLDER@(folder)

Arguments   folder   The name of the folder being deleted.

Description   Moves the folder contents to the Wastebasket and deletes the folder and all folder attributes. If folder does not exist or cannot be deleted, the macro returns FALSE. When folder is deleted, the macro returns TRUE.

See also   **MAIL_LIST_OF_FOLDERS@**

MAIL_DELETE_MESSAGE@

Deletes a message

Format MAIL_DELETE_MESSAGE@(format **inb_item** message, source)

Arguments message   The message to delete.

source   The source folder.

Description Deletes a message from a folder. Messages deleted from folders other than Outbox or Wastebasket are moved to the Wastebasket. Messages deleted from Outbox or Wastebasket are destroyed.

See also **MAIL_DELETE_SELECTED_MESSAGES@**

MAIL_DELETE_SELECTED_MESSAGES@

Deletes selected messages

Format MAIL_DELETE_SELECTED_MESSAGES@()

Description Deletes the selected messages from the current folder. The messages are placed in the Wastebasket.

See also **MAIL_DELETE_MESSAGE@**

MAIL_DELETE_SELECTED_OBJECT@

Deletes the selected message or attachment

Format MAIL_DELETE_SELECTED_OBJECT@()

Description Deletes the selected message, or, if an attachment is selected, the attached file from the Inbox.

This macro is called by the Inbox menu option File ® Delete.

See also **MAIL_DELETE_SELECTED_MESSAGES@**

MAIL_DISPLAY_ALL@

Displays all messages in an Inbox folder

Format MAIL_DISPLAY_ALL@()

Description Rewrites the current folder so that all of its messages are displayed. This macro is most often used after executing queries that tell Mail that it should only show messages that meet your queries specifications.

This macro is called by Find ® Display All.

MAIL_DLG@

Displays the Send Mail dialog box

Format MAIL_DLG@()

Description Displays the Send Mail Dialog box. MAIL_DLG@ is called by the * ® Mail menu option in all applications. This macro will determine if you are using OpenMail or Send Mail.

MAIL_EMPTY_WASTEBASKET@

Empties the Wastebasket folder

Format MAIL_EMPTY_WASTEBASKET@()

Description Empties the Wastebasket folder. If certified mail in the Wastebasket is being deleted un-read, this macro also notifies the message's sender .

MAIL_ENCRYPT_MESSAGE@

Encrypts a message

Format flag = MAIL_ENCRYPT_MESSAGE@(format **inb_item** message, folder, key)

Arguments message   The message.

folder   The source folder.

key   The encryption password.

Description Encrypts all parts of a message, except for the **envelope**. If the message cannot be encrypted, the macro returns FALSE. When the message is encrypted, the macro returns TRUE.

See also **MAIL_ENCRYPT_SELECTED_MESSAGE@**

MAIL_ENCRYPT_SELECTED_MESSAGE@

Encrypts the selected message

Format MAIL_ENCRYPT_SELECTED_MESSAGE@()

Description Encrypts the selected message.

See also **MAIL_ENCRYPT_MESSAGE@**

MAIL_FORWARD_MESSAGE@

Forwards a passed message

Format flag = MAIL_FORWARD_MESSAGE@(recipient, carbonCopyRecipient, blindCopyRecipient, subject, body, attachFiles, replyByFlag, replyText, urgentFlag, certifyFlag, outboxFlag, addRecipFlag, encryptFlag, format inb_item message)

Arguments recipient   The array of message recipients.

carbonCopyRecipient
                    The array of copy recipients.

blindCopyRecipient
                    The array of blind copy recipients.

subject   The string containing the message subject text.

body   The array of strings containing the body of the message.

attachFiles   Attached files for the message.

replyByFlag   Boolean value where TRUE means that a reply is requested.

replyText   The reply by string added to the message when replyByFlag is TRUE.

urgentFlag   Boolean value where TRUE means that the message is urgent.

certifyFlag   Boolean value where TRUE means that the message is certified.

outboxFlag	Boolean value where TRUE means save a copy of this message to the Outbox.	
addRecipFlag		
	Boolean value where TRUE means add recipients to the personal user list.	
encryptFlag	Boolean value where TRUE means encrypt the message.	
message	The forwarded message.	

Description Forwards a message to recipients. The forwarded message is appended to the message under construction, including information about the forwarded message's original sender. This macro returns TRUE if the message is successfully forwarded; otherwise, it returns FALSE.

See also **MAIL_FORWARD_SELECTED_MESSAGE@**

MAIL_FORWARD_SELECTED_MESSAGE@

Forwards the selected message

Format MAIL_FORWARD_SELECTED_MESSAGE@()

Description Forwards the selected message.

See also **MAIL_FORWARD_MESSAGE@**

MAIL_GET_BODY@

Returns the body of a message

Format bodyArray = MAIL_GET_BODY@(format **inb_item** message)

Arguments message   The passed message.

Description Returns the body of the passed message as an array of strings. If the message does not exist or an error is encountered, it returns NULL.

MAIL_GET_CC_FIELD@

Returns an array of carbon copy recipients

Format   recipientArray = MAIL_GET_CC_FIELD@(format **inb_item** message)

Arguments   message   The passed message.

Description   Returns an array that contains the carbon copy recipients from the Cc: field of the message.

MAIL_GET_CURRENT_FOLDER@

Returns the current folder's name

Format   folderName = MAIL_GET_CURRENT_FOLDER@()

Description   Returns a string containing the name of the current mail folder. Note that the folder named "Mail" is returned as Inbox. If the Inbox window is not open, NULL is returned.

MAIL_GET_FOLDER_ATTRS@

Returns folder attributes

Format   format folder_info_ info = MAIL_GET_FOLDER_ATTRS@( )

Description   Returns a format that contains a description of the currently selected folder's attributes. (The structure **folder_info_** contains the **access_data** structure.)

NULL is returned if all attributes are at their default value. Therefore, if you have not changed any values in the Folder ® Folder Attributes menu option, NULL is returned.

See also   **MAIL_WRITE_FOLDER_ATTRS@**

MAIL_GET_FROM@

Returns an array of sender names

Format   senderArray = MAIL_GET_FROM@(format **inb_item** message)

Arguments message   The message.

Description Returns an array of sender names. These are the names that appear in the From field of the message header. (The string "From " has a trailing space.)

See also **MAIL_GET_FROM_COLON@**

MAIL_GET_FROM_COLON@

Returns an array of sender names

Format senderArray = MAIL_GET_FROM_COLON@(format **inb_item** message)

Arguments message   The message.

Description Returns an array of sender names. These are the names that appear in the From: field of the message header.

See also **MAIL_GET_FROM@**

MAIL_GET_HEADER@

Returns the UNIX mail envelope

Format headerArray = MAIL_GET_HEADER@(format **inb_item** message)

Arguments message   The message.

Description Returns the UNIX mail envelope for the message as an array of strings. If message does not exist or an error is encountered, it returns NULL.

MAIL_GET_LAST_SELECTION@

Returns the most recent selection

Format selection = MAIL_GET_LAST_SELECTION@()

Description Returns the most recent selection in the current folder. It returns NULL if there is no selection.

MAIL_GET_MESSAGE_RECIPIENTS@

Returns a message's list of recipients

Format   list = MAIL_GET_MESSAGE_RECIPIENTS@(format **inb_item** message)

Arguments   message   The message.

Description   Returns a list of recipients from the message header.

MAIL_GET_MSG_LOG@

Returns a buffer containing the message log

Format   buffer = MAIL_GET_MSG_LOG@(format **inb_item** message)

Arguments   message   The message.

Description   Returns a buffer containing the message log for the message. Returns NULL if the message does not have a message log.

MAIL_GET_REPLY_TO@

Returns an array of Reply-To names

Format   replyArray = MAIL_GET_REPLY_TO@(format **inb_item** message)

Arguments   message   The message.

Description   Returns an array of names from the Reply-To: field of the message header.

MAIL_GET_SELECTED_ATTACHMENT@

Returns the selected attachment's path

Format   sourceFile = MAIL_GET_SELECTED_ATTACHMENT@()

Description   Retrieves the path of the selected attached file.

MAIL_GET_SELECTED_MESSAGES@

Returns a list of selected messages

Format   format arrayof inb_item msgs = MAIL_GET_SELECTED_MESSAGES@()

Description   Retrieves a list of selected messages.

MAIL_GET_TO_FIELD@

Returns an array of To recipients

Format   recipientArray = MAIL_GET_TO_FIELD@format **inb_item** message)

Arguments   message   The message.

Description   Returns an array of To recipients from message's To: field.

MAIL_HAS_ATTACHMENTS@

Indicates if a message has attachments

Format   flag = MAIL_HAS_ATTACHMENTS@(format **inb_item** message)

Arguments   message   The message.

Description   Returns TRUE if message has attachments; FALSE is returned if message does not have attachments.

See also   **MAIL_ATTACHED_FILES@**
MAIL_GET_SELECTED_ATTACHMENT@
MAIL_SAVE_ATTACHMENT@

MAIL_IMPORT_FOLDER@

Imports the mbox format file into a folder

Format   flag = MAIL_IMPORT_FOLDER@(folder, mailbox, passFlag)

Arguments folder   The target folder.

mailbox   The mbox format file.

passFlag   Boolean value where TRUE means use autodisposition rules to place the imported files into folders.

Description Imports mailbox, which is an operating system file, into the Applixware mail system. If passFlag is TRUE, autodisposition rules are applied to the messages within mailbox. If passFlag is FALSE, the messages are imported into the current folder.

This macro returns TRUE if messages are moved to folders. Otherwise, it returns FALSE.

MAIL_IS_CERTIFIED@

Indicates if a message is certified

Format flag = MAIL_IS_CERTIFIED@(format **inb_item** message)

Arguments message   The message.

Description Returns TRUE if message is certified; it returns FALSE if message is not certified.

MAIL_IS_DOCUMENT@

Indicates if a message is an Applixware document

Format flag = MAIL_IS_DOCUMENT@(format **inb_item** message)

Arguments message   The message.

Description Returns TRUE if message is an Applixware document; otherwise it returns FALSE.

MAIL_IS_ENCRYPTED@

Indicates if a message is encrypted

Format flag = MAIL_IS_ENCRYPTED@(format **inb_item** message)

Arguments message   The message.

Description Returns TRUE if message is encrypted; otherwise it returns FALSE.

MAIL_IS_SIGNED@

Indicates if a message is signed

Format   flag = MAIL_IS_SIGNED@(format **inb_item** message)

Arguments   message   The message.

Description   Returns TRUE if message was signed by the sender; otherwise it returns FALSE.

MAIL_IS_TAGGED@

Indicates if a message is tagged

Format   flag = MAIL_IS_TAGGED@(format **inb_item** message)

Arguments   message   The message.

Description   Returns TRUE if message is tagged, otherwise it returns FALSE. A tagged message is displayed in ***bold italic*** text in the message list area. (Messages are tagged to mark a mail message for further operation such as moving, copying, deleting, encrypting, and so on.)

MAIL_IS_UNREAD@

Indicates if message was read

Format   flag = MAIL_IS_UNREAD@(format **inb_item** message)

Arguments   message   The message.

Description   Returns TRUE if the message has been read; otherwise, it returns FALSE.

A "read" message is a message that was opened. Note that this is asking if a *received* message was read; it is not asking if a *sent* message was read.

MAIL_IS_URGENT@

Indicates if a message is urgent

Format flag = MAIL_IS_URGENT@(format **inb_item** message)

Arguments message   The message.

Description Returns TRUE if message is urgent, otherwise it returns FALSE.

MAIL_KEYWORD_QUERY@

Queries messages by keyword in the Inbox

Format MAIL_KEYWORD_QUERY@(type, keyword, matchCaseFlag)

Arguments type   One of the following type of keyword queries:

MAIL#BODY_   Keyword in message body
MAIL#COMMENT_   Keyword in comment
MAIL#DATE_   Keyword in date
MAIL#SENDER_   Keyword in sender
MAIL#SUBJECT_   Keyword in subject

The header file containing the DEFINE statements for the type values is mail_.am.

keyword   The keyword for the query.

matchCaseFlag
Boolean value where TRUE means that a case-sensitive match is required; FALSE indicates that the case of keyword does not matter.

Description Queries messages by keyword in the Inbox, searching in the area of the message determined by stateFlag. The macro is called by the Find ® By Keyword option. The messages in the current folder that meet the keyword query are displayed.

A query acts on the messages that are displayed in the current folder. A second query acts on the results of a first query, not on all the messages in the folder.

See also **MAIL_COMBINE_QUERY@**

MAIL_LIST_OF_FOLDERS@

Returns a list of folders

Format foldersArray = MAIL_LIST_OF_FOLDERS@()

Description Returns an array that contains the names of folders located in the user's axMail directory. It also contains the names of any subscribed shared folders owned by other users.

MAIL_LIST_OF_STUFF@

Returns a list of files associated with a message

Format fileArray = MAIL_LIST_OF_STUFF@(format **inb_item** message)

Arguments message   The message.

Description Returns an array whose contents contain the names of files associated with message. The name of each file includes the full Mailbox path name.

MAIL_MESSAGE_REPLY@

Sends a reply to the message

Format flag = MAIL_MESSAGE_REPLY@(format **inb_item** message, unused, carbonCopyList, blindCopyList, subject, messageBody, includeOriginalFlag, urgentFlag, certifyFlag, outboxFlag, replyByFlag, replyText)

Arguments message   The message.

unused   Not Used.

carbonCopyList
            An array of carbon copy recipient names.

blindCopyList
            An array of blind copy recipient names.

subject   A string containing the reply message's subject.

messageBody
            An array of strings containing the reply message's body.

includeOriginalFlag
  Boolean value where TRUE means include the original message in the re-
  ply.

urgentFlag   Boolean value where TRUE means that the message is urgent.

certifyFlag   Boolean value where TRUE means that the message is certified.

outboxFlag   Boolean value where TRUE means save a copy of this message to the
              Outbox.

replyByFlag   Boolean value where TRUE means that a reply is requested.

replyText   The reply by string added to the message when replyByFlag is TRUE.

Description Sends a reply to the message, using the recipient, carbon copy, and blind copy recipient
lists from the passed message. The macro returns TRUE on successful reply, otherwise
returns FALSE.

See also **MAIL_REPLY_MESSAGE@**

MAIL_MOVE_MESSAGE@

Moves a message to a folder

Format MAIL_MOVE_MESSAGE@(format **inb_item** message, source, destination)

Arguments message   The message.

source   The source folder.

destination The destination folder.

Description Moves a message from the source folder to the destination folder.

See also **MAIL_MOVE_SELECTED_MESSAGES@**

MAIL_MOVE_SELECTED_MESSAGES@

Moves selected messages to a folder

Format MAIL_MOVE_SELECTED_MESSAGES@(destination)

Arguments destination   The destination folder.

Description Moves the selected messages to the target folder.

See also **MAIL_MOVE_MESSAGE@**

MAIL_NEW_MESSAGE_COUNT@

Returns the number of unread messages

Format messages = MAIL_NEW_MESSAGE_COUNT@(folder)

Arguments folder   The folder.

Description Returns the number of unread messages in folder.

MAIL_OPEN_SELECTED_FOLDER@

Opens the selected mail folder

Format flag = MAIL_OPEN_SELECTED_FOLDER@()

Description Opens the selected mail folder, returning TRUE if it can open the folder; FALSE is returned if the folder cannot be opened

---

MAIL_PRINT_MESSAGES@

Prints the mail message

Format flag = MAIL_PRINT_MESSAGES@(format arrayof **inb_item** message, printer, colorFlag, copies, bannerFlag, collateFlag, attachFlag, envelopeFlag, commentFlag, printFile, backgrndFlag, class)

Arguments
message	The array of messages.
printer	The destination printer.
colorFlag	Boolean value where TRUE means print message in color.
copies	The number of copies to print.
bannerFlag	Boolean value where TRUE means "use banner page".
collateFlag	Boolean value where TRUE means collate printed copies".
attachFlag	Boolean value where TRUE means "print attached files".

envelopeFlag

        Boolean value where TRUE means "print envelope information".

commentFlag

        Boolean value where TRUE means "print message comments".

printFile   The name of the print file.

backgrndFlag

        Boolean value where TRUE means "print message in background".

class   The string representing the printer class of the selected printer. The valid printer class values are:

PostScript	PostScript printer
PCL5	PCL5 printer

Description Prints the mail message. It returns TRUE if the message is sent to the printer; otherwise it returns FALSE.

See also **MAIL_PRINT_SELECTED_MESSAGES@**

MAIL_PRINT_SELECTED_MESSAGES@

Prints the selected messages

Format flag = MAIL_PRINT_SELECTED_MESSAGES@(printer, colorFlag, copies, bannerFlag,
        collateFlag, attachFlag,
        envelopeFlag, commentFlag, printFiles,
        backgrndFlag, class)

Arguments printer   The destination printer.

colorFlag   Boolean value where TRUE means print messages in color.

copies   The number of copies to print.

bannerFlag Boolean value where TRUE means "use banner page".

collateFlag Boolean value where TRUE means "collate printed copies".

attachFlag Boolean value where TRUE means "print attached files".

envelopeFlag

        Boolean value where TRUE means "print envelope information".

commentFlag

        Boolean value where TRUE means "print messages comments".

printFiles   The names of the print files.

backgrndFlag

Boolean value where TRUE means "print messages in background".

class
The string representing the printer class of the selected printer. The valid printer class values are:

PostScript   PostScript printer
PCL5   PCL5 printer

Description Prints the selected Inbox messages. It returns TRUE if the message is sent to the printer; otherwise it returns FALSE.

See also **MAIL_PRINT_MESSAGES@**

MAIL_PRINT_SELECTED_OBJECT@

MAIL_PRINT_SELECTED_OBJECT@

Prints the selected message or attachment

Format MAIL_PRINT_SELECTED_OBJECT@()

Description Prints the selected object. The object can be the current message or an attachment to the current message.

This macro is called by the Inbox menu option File ® Print.

MAIL_READ_FOLDER@

Returns the contents of a folder

Format format arrayof inb_item messages = MAIL_READ_FOLDER@(folder)

Arguments folder   The folder.

Description Reads folder, returning the folder's contents as an array of **inb_item** messages. This format is defined in mail_.am.

MAIL_READ_SELECTED_MESSAGE@

Opens a selected message

Format MAIL_READ_SELECTED_MESSAGE@()

Description Opens the selected Inbox message.

MAIL_READ_SELECTED_OBJECT@

Opens selected message or attachment

Format MAIL_READ_SELECTED_OBJECT@()

Description Opens the selected message, or, if an attachment is selected, the attached file from the Inbox.

This macro is called by the Inbox menu option File ® Read.

MAIL_RECIPIENT_PREFERENCES@

Returns recipient preferences

Format format mail_recip_prefs_ preferences = MAIL_RECIPIENT_PREFERENCES@
            (recipient)

Arguments recipient   The recipient.

Description Returns a structure containing the recipient's preferences. This structure is contained within mail_.am and is as follows:

format mail_recip_prefs_
        username,   'recipient name
        wp_pref,   'wp doc format pref
        gr_pref,   'gr doc format pref
        ss_pref,   'ss doc format pref
        mac_user,   `true or false (for mac hdr)
        fax_phone,   'phone number as dialed
        fax_logname,   'full name for header page
        fax_send_time,   `deferred send time or 0
        mime,   'true or false (wants mime)
        dd_pref   'dd doc format pref
        dd_pref_func,   'dd conversion func
        unused1,
        unused2,
        unused3,
        unused4

This macro does not resolve alias names. In this way, a person can have more than one preference for the same person.

If you use an asterisk (*) as the recipient name, you will receive the default preference setting.

MAIL_RECORD_MACRO@

Records a macro

Format   MAIL_RECORD_MACRO@()

Description   Records a macro within the Send Mail dialog box.

MAIL_REMOVE_ATTACHMENT@

Removes attachment from a message being created

Format   flag = MAIL_REMOVE_ATTACHMENT@(filename)

Arguments   filename   The name of the file being deleted.

Description   Removes an attachment from the message being created.

When you attach a file to a message, you are copying the attachment file into memory. Removing the attachment means that you are deleting it from memory, not deleting it from disk. That is, the original file is not deleted¾just the copy.

MAIL_REMOVE_BC_RECIPIENT@

Removes blind copy recipient name

Format   flag = MAIL_REMOVE_BC_RECIPIENT@(recipient)

Arguments   recipient   The recipient name.

Description   Removes a recipient name from the blind copy recipient list in the current message. This macro returns TRUE if the name is removed; otherwise, it returns FALSE.

See also   **MAIL_REMOVE_CC_RECIPIENT@**
MAIL_REMOVE_TO_RECIPIENT@

MAIL_REMOVE_CC_RECIPIENT@

Removes carbon copy recipient name

Format flag = MAIL_REMOVE_CC_RECIPIENT@(recipient)

Arguments recipient   The recipient name.

Description Removes a recipient name from the carbon copy recipient list in the current message. This macro returns TRUE if the name is removed; otherwise, it returns FALSE.

See also **MAIL_REMOVE_BC_RECIPIENT@**
MAIL_REMOVE_TO_RECIPIENT@

MAIL_REMOVE_TO_RECIPIENT@

Removes original recipient name

Format flag = MAIL_REMOVE_TO_RECIPIENT@(recipient)

Arguments recipient   The recipient name.

Description Removes the recipient name from the original recipient list in the current message. This macro returns TRUE if the name is removed; otherwise, it returns FALSE.

See also **MAIL_REMOVE_BC_RECIPIENT@**
MAIL_REMOVE_CC_RECIPIENT@

MAIL_RENAME_FOLDER@

Renames a mail folder

Format flag = MAIL_RENAME_FOLDER@(oldName, newName)

Arguments oldName   The original name of the folder.

newName   The new folder name.

Description Renames a mail folder. The macro returns TRUE if the folder is renamed; otherwise, it returns FALSE.

MAIL_REPLY_MESSAGE@

Replies to a message

Format flag = MAIL_REPLY_MESSAGE@(recipient, carbonCopyRecipient, blindCopyRecipient, subject, body, attachFiles, replyByFlag, replyText, urgentFlag, certifyFlag, outboxFlag, addRecipFlag, encryptFlag, format **inb_item** message)

Arguments recipient   An array of message recipients.

carbonCopyRecipient
     An of carbon copy recipients.

blindCopyRecipient
     An of blind copy recipients.

subject   A string containing the message subject text.

body   An array of strings containing the body of the message under construction.

attachFiles   Attached files for the message under construction.

replyByFlag   Boolean value where TRUE means that a reply is requested.

replyText   The reply by string added to the message when replybyFlag is TRUE.

urgentFlag   Boolean value where TRUE means that the message is urgent.

certifyFlag   Boolean value where TRUE means that the message is certified.

outboxFlag   Boolean value where TRUE means save a copy of this message to the Outbox.

addRecipFlag
     Boolean value where TRUE means copy recipients to the personal user list.

encryptFlag   Boolean value where TRUE means encrypt the message.

message   The reply message.

Description Replies to a message. The original message is appended to the message under construction, including information about the original message's original sender. Returns TRUE on successful reply, otherwise returns FALSE.

See also **MAIL_MESSAGE_REPLY@**

MAIL_RESOLVE_ALIAS@

Returns alias value

Format alias = MAIL_RESOLVE_ALIAS@(userName)

Arguments userName   The name of the user.

Description Returns the alias defined for a userName. If no alias was defined for userName, user-Name is returned. Because a name is always returned, you are guaranteed to have a name that can be used to send a message.

MAIL_RESTORE_CONTEXT@

Restores a Send Mail message

Format MAIL_RESTORE_CONTEXT@()

Description Restores the saved context of a Send Mail message. That is, this macro imports a message that you have saved to your directory using **MAIL_SAVE_CONTEXT@**.

After you use this macro, you can mail the message or use Save Context to save the message again. The file remains in the directory where you saved it until you manually delete it.

MAIL_SAVE_ATTACHMENT@

Saves an attached file as a separate file in the file system

Format flag = MAIL_SAVE_ATTACHMENT@(originalFile, newFile, permission, forceFlag)

Arguments originalFile   The name of an attached file.

newFile   The new file name.

permission   The absolute octal mode string value associated with file permission. This value sets the read, or read and write permissions to Group or Others. Values you can use are:

640   Group read-only
644   Group and Other read-only

	664	Group read and write, Other read-only
	666	Group and Other read and write
forceFlag		Boolean value where TRUE means force an overwrite on an existing new-File with the attached file.

Description Saves the attached file as a separate file in the file system. This macro returns TRUE if the attachment is saved; otherwise, it returns FALSE.

See also **MAIL_SAVE_SELECTED_OBJECT@**

MAIL_SAVE_CONTEXT@

Saves the context of a Send Mail message

Format MAIL_SAVE_CONTEXT@(fileName)

Arguments filename   The message context file name.

Description Save a complete or partially composed message to your UNIX directory in ASCII format pending future use. In addition to saving an incomplete message, pending messages can be useful as template messages, for example, for a weekly status report.

For information on restoring a Send Mail message context with a saved message context, see **MAIL_RESTORE_CONTEXT@**.

MAIL_SAVE_FOLDER@

Writes folder contents to an mbox file

Format flag = MAIL_SAVE_FOLDER@(folder, fileName, mode, saveAttachFlag, permission)

Arguments folder   The folder.

fileName   The new mailbox file name.

mode   The save mode of the file.

saveAttachFlag
        Boolean value where TRUE means write attached files into fileName as well as write the messages.

permission   The absolute octal mode string value associated with file permission. Assigns read, or read and write permissions to Group or Others. The possible values are:

        640   Group read-only

644	Group and Other read-only
664	Group read and write, Other read-only
666	Group and Other read and write

Description   Saves folder's contents to a file in mbox format. (An mbox file is a UNIX mail file.) This macro returns TRUE if the information is saved to fileName; otherwise, it returns FALSE.

See also   MAIL_SAVE_MESSAGES@

MAIL_SAVE_MESSAGES@

Writes messages to an mbox file

Format   flag = MAIL_SAVE_MESSAGES@(format arrayof **inb_item** messages, fileName, mode, saveAttachFlag, permission)

Arguments   messages   The array of messages.

fileName   The new mailbox file name.

mode   The save mode of the file.

saveAttachFlag
Boolean value where TRUE means write attached files into fileName as well as write the messages.

permission   The value associated with file permission. Assigns read, or read and write permissions to Group or Others. The possible values are:

640	Group read-only
644	Group and Other read-only
664	Group read and write, Other read-only
666	Group and Other read and write

Description   Saves messages' contents to a file in mbox format. (An mbox file is a UNIX mail file.) This macro returns TRUE if the information is saved to fileName; otherwise, it returns FALSE.

See also   MAIL_SAVE_SELECTED_OBJECT@

MAIL_SAVE_FOLDER@

MAIL_SAVE_SELECTED_OBJECT@

Saves the selected message or attachment

Format MAIL_SAVE_SELECTED_OBJECT@()

Description Saves the selected message, or, if an attachment is selected, the attached file from the Inbox. This macro is called by the Inbox menu option File ® Save As.

MAIL_SELECT_ALL@

Selects all messages in current folder

Format MAIL_SELECT_ALL@()

Description Selects all messages in the current Inbox folder.

MAIL_SELECT_MESSAGES@

Selects specified messages

Format flag = MAIL_SELECT_MESSAGES@(messageList)

Arguments messageList
An array of message index values.

Description Selects the messages in the current folder specified by the index values. Index values are 1-based; that is, the first message is message 1, not message 0. Returns TRUE if successful; otherwise, it returns FALSE.

MAIL_SENDER_QUERY@

Queries Inbox based on a sender/recipient list

Format MAIL_SENDER_QUERY@(senderList)

Arguments senderList   The sender/recipient.

Description Queries the current Inbox display based on the sender/recipient list.

A query acts on the messages that are displayed in the current folder. A second query acts on the results of a first query, not on all the messages in the folder.

See also **MAIL_COMBINE_QUERY@**

MAIL_SEND_DOC@

Sends documents

Format flag = MAIL_SEND_DOC@(recipient, carbonCopyRecipient, blindCopyRecipient, subject, body, applicationId, documents, format, replyByFlag, replyText, urgentFlag, certifyFlag, outboxFlag, addRecipFlag, localizeFlag, encryptFlag)

Arguments recipient   An array of message recipients.

carbonCopyRecipient
               An array of carbon copy recipients.

blindCopyRecipient
               An array of blind copy recipients.

subject   A string containing the message subject text.

body   An array of strings containing the body of the message.

applicationId The document's application ID, which is used to determine what conversion formats to offer. Application IDs are defined in app_ids_.am.

documents   A two dimensional array of documents. Item 0 of the array contains the copy of the document that is actually sent. Item 1 of the array contains the full path name of the original file.

format   The application document format.

replyByFlag   Boolean value where TRUE means that a reply is requested.

replyText   The reply by string added to the message when replyByFlag is TRUE.

urgentFlag   Boolean value where TRUE means that the message is urgent.

certifyFlag   Boolean value where TRUE means that the message is certified.

outboxFlag   Boolean value where TRUE means "save a copy of this message to the Outbox".

addRecipFlag
               Boolean value where TRUE means "copy recipients to the personal user list".

localizeFlag   Boolean value where TRUE means "localize linked objects in the document".

encryptFlag   Boolean value where TRUE means encrypt message.

Description   Sends an Applixware document through Applixware Mail with a mail message. You can send a document in any format accepted by other software for which you have Applixware filters installed on your system. If the document is sent successfully, a TRUE is returned; otherwise, it returns FALSE.

MAIL_SEND_MESSAGE@

Sends a message

Format   flag = MAIL_SEND_MESSAGE@(recipient, carbonCopyRecipient, blindCopyRecipient, subject, body, attachFiles, replyByFlag, replyText, urgentFlag, certifyFlag, outboxFlag, addRecipFlag, encryptFlag, key)

Arguments   recipient   An array of message recipients.

carbonCopyRecipient
An array of carbon copy recipients.

blindCopyRecipient
An array of blind copy recipients.

subject   A string containing the message subject text.

body   An array of strings containing the body of the message under construction.

attachFiles   Attached files for the message under construction.

replyByFlag   Boolean value where TRUE means that a reply is requested.

replyText   The reply by string added to the message when replyByFlag is TRUE.

urgentFlag   Boolean value where TRUE means that the message is urgent.

certifyFlag   Boolean value where TRUE means that the message is certified.

outboxFlag   Boolean value where TRUE means "save a copy of this message to the Outbox".

addRecipFlag
Boolean value where TRUE means "copy recipients to the personal user list".

encryptFlag   Boolean value where TRUE means "encrypt message".

key   The encryption key.

Description   Sends a message from the Send Mail dialog box. If the document is sent successfully, a TRUE is returned; otherwise, it returns FALSE.

MAIL_SHELL_COMMAND@

Executes a shell command

Format resultArray = MAIL_SHELL_COMMAND@(command)

Arguments command A command that can be executed by the shell.

Description Executes a shell command, returning the exit code and command result.

This command uses the **RUN_PROGRAM@** macro. Unlike running RUN_PRO-GRAM@, running this macro will throw an error if a problem occurs.

MAIL_SHOW_NUMBERS@

Controls the display of message numbers in the Inbox

Format MAIL_SHOW_NUMBERS@(toggle)

Arguments toggle Boolean value where TRUE means display message numbers.

Description Controls the display of message numbers in the Inbox with the Boolean value.

See also **MAIL_TOGGLE_NUMBERS@**

MAIL_SHOW_SHARED_FOLDERS@

Controls display of shared folders in the Inbox

Format MAIL_SHOW_SHARED_FOLDERS@(flag)

Arguments flag Boolean value where TRUE means display shared folders.

Description Controls the display of shared folders in the Inbox with a Boolean value.

MAIL_SIGNATURE@

Creates or modifies a signature for the user

Format MAIL_SIGNATURE@()

Description Prompts the user to create a signature if one does not exist. If a signature exists for the user, the macro allows the user to edit the signature.

MAIL_STATUS_QUERY@

Queries current folder messages by status

Format MAIL_STATUS_QUERY@(unreadFlag, urgentFlag, taggedFlag, attachFlag, documentFlag, audioFlag)

Arguments unreadFlag   Boolean value where TRUE means find unread messages.

urgentFlag   Boolean value where TRUE means find urgent messages.

taggedFlag   Boolean value where TRUE means find tagged messages.

attachFlag   Boolean value where TRUE means find messages with attached files.

documentFlag
Boolean value where TRUE means find messages that are mailed Applixware documents.

audioFlag   Boolean value where TRUE means find messages that contain audio components.

Description Queries the current folder using status values and displays the messages that have the status flags set.

A query acts on the messages that are displayed in the current folder. A second query acts on the results of a first query, not on all the messages in the folder.

See also **MAIL_COMBINE_QUERY@**

MAIL_TAG_MESSAGE@

Tags the current folder message

Format MAIL_TAG_MESSAGE@(format **inb_item** message, source)

Arguments message   The message.

source   The source folder.

Description Tags the current folder message and writes the file to the source folder. A tagged message is displayed in bold italic text in the message list area.

Messages are tagged to mark a mail message for further operation such as moving, copying, deleting, encrypting, and so on.

See also **MAIL_TAG_SELECTED_MESSAGES@**

        MAIL_UNTAG_MESSAGE@

MAIL_TAG_SELECTED_MESSAGES@

Tags selected messages

Format MAIL_TAG_SELECTED_MESSAGES@()

Description Tags the selected messages in the Inbox. A tagged message is displayed in bold italic text in the message list area.

Messages are tagged to mark a mail message for further operation such as moving, copying, deleting, encrypting, and so on.

See also **MAIL_TAG_MESSAGE@**

        MAIL_UNTAG_SELECTED_MESSAGES@

MAIL_TOGGLE_NUMBERS@

Toggles the display of message numbers

Format MAIL_TOGGLE_NUMBERS@()

Description Toggles the display of message numbers in the Inbox.

See also **MAIL_SHOW_NUMBERS@**

MAIL_TOGGLE_SHARED@

Toggles the display of shared folders in the Inbox

Format MAIL_TOGGLE_SHARED@()

Description Toggles the display of shared folders in the Inbox.

MAIL_UNTAG_MESSAGE@

Untags a message

Format   MAIL_UNTAG_MESSAGE@(format **inb_item** message, source)

Arguments   message   A message.

source   A source folder.

Description   Untags a message.

See also   **MAIL_TAG_MESSAGE@**
MAIL_UNTAG_SELECTED_MESSAGES@

MAIL_UNTAG_SELECTED_MESSAGES@

Untags selected messages

Format   MAIL_UNTAG_SELECTED_MESSAGES@()

Description   Untags the selected messages.

See also   **MAIL_TAG_SELECTED_MESSAGES@**
MAIL_UNTAG_MESSAGE@

MAIL_WRITE_FOLDER@

Writes an ELF array of messages to a folder

Format   MAIL_WRITE_FOLDER@(folder, format **inb_item** messages)

Arguments   folder   The destination folder.

messages   The mail messages.

Description   Writes messages to the destination folder.

MAIL_RECOVER@

Recovers lost mail

Format MAIL_RECOVER@()

Description Recovers lost mail, mail for which contents are found but is not referenced in a mail folder. Recovered messages are restored to a folder named Recovery.

MAIL_REHASH_GLOBAL_USERLIST@

Rebuilds global user list on next display

Format MAIL_REHASH_GLOBAL_USERLIST@()

Description Forces a rebuild of the global user list on the next display. The global user list is regenerated every time you start Applixware. To generate the global user list without logging out and back in, run this macro.

By default, this macro generates a list of users from the /etc/passwd file. You can specify a macro to retrieve a different file of user names by naming a macro in the Mail Preferences ® Functions dialog box.

MAIL_SIZE_OF_FOLDER@

Returns the number of messages in a folder

Format number = MAIL_SIZE_OF_FOLDER@(folder)

Arguments folder   A folder.

Description Returns the number of messages in a local folder.

MAIL_UPDATE_MSG_LOG@

Adds log data to a message

Format flag = MAIL_UPDATE_MSG_LOG@(format **inb_item** message, text)

Arguments message   A message.

text   The text string of message log data.

Description Adds text to the Inbox message.

MAIL_WRITE_FOLDER_ATTRS@

Writes a structure of folder attributes

Format   flag = MAIL_WRITE_FOLDER_ATTRS@(format **folder_info** attributes)

Arguments attributes   The structure of folder attributes.

Description Writes a structure of folder attributes. Returns TRUE on success, otherwise returns FALSE.

See also **MAIL_GET_FOLDER_ATTRS@**

CREATE_MAIL_SS_COPY@

Writes temporary copy of file into mail temporary directory

Format   mailNameArray = CREATE_MAIL_SS_COPY@()

Description Writes a copy of the current Spreadsheets document into Applixware Mail's temporary directory.

This macro is one of the macros executed by Spreadsheets when you choose File ® Send. This temporary copy contains all of your current changes (even if the changes have not yet been saved to disk). Ordinarily, this temporary file is deleted after the file is mailed.

The following two-element array is returned:

array[0,0]   The name of the temporary file.

array[0,1]   The file's original file name.

CREATE_MAIL_WP_COPY@

Writes temporary copy of current file in mail temporary directory

Format mailNamesArray = CREATE_MAIL_WP_COPY@()

Description Writes a copy of the current Words document into Applixware Mail's temporary directory.

This macro is one of the macros executed by Spreadsheets when you choose File ® Send. This temporary copy contains all of your current changes (even if the changes have not yet been saved to disk). Ordinarily, this temporary file is deleted after the file is mailed.

The following two-element array is returned:

array[0,0]   The name of the temporary file.

array[0,1]   The file's original file name.

DISABLE_MAIL@

Turns off the automatic import of system mail

Format DISABLE_MAIL@()

Description Sets the automatic import of system mail to FALSE. In addition, it disables the New Mail Notifiers, turns off notifier beeps, and stops the mail polling process.

See also ENABLE_MAIL@

ENABLE_MAIL@

Turns on the automatic import of system mail

Format ENABLE_MAIL@(autoimpFlag)

Arguments autoimpFlag The automatic import option.

Description Enables the automatic import of system mail. If autoimpFlag is set to TRUE, the New Mail Notifiers, mail polling process, and single notifier beep are enabled. If autoimpFlag is FALSE, the macro disables the New Mail Notifiers, turns off notifier beeps, and stops the mail polling process.

See also DISABLE_MAIL@

INB_ADD_SENDER@

Add message sender to personal alias list

Format INB_ADD_SENDER@()

Description Adds the name of the person who sent the current message to the user's personal alias list. The Add Sender to Personal List dialog box is invoked.

To specify an optional alias for the sender, turn on the **Alias** option and type the alias to use in your personal list.

To add the sender's name or alias, click OK. Mail checks for duplicates before adding the name to your list.

This is a User Interface macro.

See the List menu options in the **Send Mail** dialog box for more information about personal lists.

INB_COPY@

Copies mail text to the clipboard

Format INB_COPY@( )

Description Copies a message's text or a selected part of it and copies it into the Applixware clipboard. The text is stored on the clipboard until another selection has been copied; you can paste the clipboard contents as many times as you want until then.

Use Edit ® **Paste** in any other application window to insert the copied text into the application's window.

INB_EMPTY_TRASH@

Empties the Inbox wastebasket

Format INB_EMPTY_TRASH@()

Description Permanently removes messages from your Inbox. Not only are they deleted from the Wastebasket folder, they are deleted from disk.

INB_PLAY_AUDIO@

Plays an audio message

Format INB_PLAY_AUDIO@()

Description Plays an audio message sent with a mail message.

The audio button will have a border around it if it is available, and is only available if you are running on a system that supports audio, or are running off a server which does. If your server supports audio but your terminal does not, the audio file will play on your server. If you are not sure where the audio file will play, ask your system administrator before you possibly torment the user who is sitting on the server.

See Attachments ® "Audio" in the Send Mail dialog box for information on sending a message with an attached audio file.

INB_QUIT@

Exits from Inbox

Format INB_QUIT@()

Description Closes the Inbox. Normal closing operations such as writing profile information and emptying of the wastebasket will occur.

INB_READ_UMAIL@

Imports system mail

Format INB_READ_UMAIL@()

Description Tells Mail that it should import mail from the system mailbox into Applixware mail. As this process occurs in the background, the Inbox can be iconified when this macro executes.

INBOX_DLG@

Displays the Inbox dialog box

Format   INBOX_DLG@([folder,[format inb_item messageList]])

Arguments   folder   The optional string containing the name of the folder to open. If folder is not supplied, the macro opens the Inbox folder when the Inbox dialog box is displayed.

           messageList The optional array of message index values for the messages to select. Index values are zero-based.

Description   Displays the Inbox dialog box. If folder is supplied, Inbox displays with folder open. By default Inbox displays with the Inbox folder open. If messageList is supplied, the messages corresponding to the passed index values are selected.

INB_RECORD_MACRO@

Records a macro

Format   INB_RECORD_MACRO@()

Description   Records a macro within the Inbox dialog box. This macro is called by the Inbox menu option * ® Record Macro.

1. This is an Applixware0 macro. Refer to **previous macros** for the equivalent macro that was available in earlier versions of Applix*ware* and/or Aster*x.

INBOX_SHOW_ENVELOPE@

Checks envelope status and displays indicated portion

Format   INBOX_SHOW_ENVELOPE@()

Description   If this option is set, displays the messages envelope. The envelope contains the message header, including routing information, recipients, and subject of the message. By default, the envelope is not displayed.

The Utilities ® **Headers** command on the Inbox menu lets you show or hide fields in the envelope.

INBOX_SORT_ALPHARCP@

Sorts mail by recipient

Format INBOX_SORT_ALPHARCP@()

Description Sorts messages in the current Inbox folder alphabetically by recipient name.

See also **INBOX_SORT_ALPHASUBJ@**
INBOX_SORT_ALPHASEND@
INBOX_SORT_DTASC@
INBOX_SORT_DTDSC@
INBOX_SORT_GROUP@
INBOX_SORT_NEW@

INBOX_SORT_ALPHASEND@

Sorts messages by sender name

Format INBOX_SORT_ALPHASEND@()

Description Sorts messages in the current Inbox folder alphabetically by sender name.
This macro is called by the Inbox menu option View ® By Sender.

See also **INBOX_SORT_ALPHASUBJ@**
INBOX_SORT_ALPHARCP@
INBOX_SORT_DTASC@
INBOX_SORT_DTDSC@
INBOX_SORT_GROUP@
INBOX_SORT_NEW@

INBOX_SORT_ALPHASUBJ@

Sorts messages by subject text

Format INBOX_SORT_ALPHASUBJ@()

Description Sorts messages in the current Inbox folder alphabetically by subject text.

This macro is called by the Inbox menu option View ® By Subject.

See also **INBOX_SORT_ALPHARCP@**

INBOX_SORT_ALPHASEND@

INBOX_SORT_DTASC@

INBOX_SORT_DTDSC@

INBOX_SORT_GROUP@

INBOX_SORT_NEW@

INBOX_SORT_DTASC@

Sorts messages in ascending date order

Format INBOX_SORT_DTASC@()

Description Sorts messages in the current Inbox folder by received date in ascending order (oldest to most recent).

This macro is called by the Inbox menu option View ® Date Ascending.

See also **INBOX_SORT_ALPHARCP@**

INBOX_SORT_ALPHASEND@

INBOX_SORT_ALPHASUBJ@

INBOX_SORT_DTDSC@

INBOX_SORT_GROUP@

INBOX_SORT_NEW@

INBOX_SORT_DTDSC@

Sorts messages in descending date order

Format INBOX_SORT_DTDSC@()

Description Sorts messages in the current Inbox folder by received date in descending order (most recent to oldest).

This macro is called by the Inbox menu option View ® Date Descending.

See also **INBOX_SORT_ALPHARCP@**

INBOX_SORT_ALPHASEND@

INBOX_SORT_ALPHASUBJ@

INBOX_SORT_DTASC@

INBOX_SORT_GROUP@

INBOX_SORT_NEW@

INBOX_SORT_GROUP@

Places Urgent messages at top of folder

Format INBOX_SORT_GROUP@

Description Places Urgent messages at the top of the displayed Inbox folder.

This macro is called by the Inbox menu option View ® Urgent Mail First.

See also **INBOX_SORT_ALPHARCP@**

INBOX_SORT_ALPHASEND@

INBOX_SORT_ALPHASUBJ@

INBOX_SORT_DTASC@

INBOX_SORT_DTDSC@

INBOX_SORT_NEW@

INBOX_SORT_NEW@

Places unread messages at top of folder

Format INBOX_SORT_NEW@()

Description Places unread messages at the top of the current Inbox folder. This macro is called by the Inbox menu option View ® Unread Mail First.

See also **INBOX_SORT_ALPHARCP@**

INBOX_SORT_ALPHASEND@

INBOX_SORT_ALPHASUBJ@

INBOX_SORT_DTASC@

INBOX_SORT_DTDSC@

INBOX_SORT_GROUP@

INBOX_SORT_NONE@

Removes all special sorting off in the inbox

Format   INBOX_SORT_NONE@()

Description   Returns the Inbox to the default sort order. Messages appear sorted by their date and time stamp, and all sort buttons in the Inbox View menu are turned off.

See also   **INBOX_SORT_ALPHARCP@**

            INBOX_SORT_ALPHASEND@

            INBOX_SORT_ALPHASUBJ@

            INBOX_SORT_DTASC@

            INBOX_SORT_DTDSC@

            INBOX_SORT_GROUP@

Autodisposition

Autodisposition Rules allow you to automatically perform actions on mail messages as you receive them.   If you receive large quantities of mail each day, or on certain days, you can save time by writing and enabling Autodisposition Rules to file messages, send replies and documents, forward messages, and anything else you need to do.

For example, you may manage a large group of people who must file weekly status reports. Although you want to read all of them, you prefer to read them all at the same time. You can write an Autodisposition Rule which looks for mail from particular senders with a particular word or words in the subject, and/or a certain attachment, and files them in a folder called "Status."

Even if you have one or more of the New Mail notifiers turned on, you may prefer to not to be notified each time a status report comes in. In that case, you can change the folder attributes of a particular folder to suppress notification of new mail (except for Mail, Wastebasket and Outbox). See Folder ® **Folder Attributes** for more information.

Autodisposition Rules can also be applied to messages you import in mbox format.

For more information, select one of the following:
Autodisposition Examples
How Autodisposition Works
ELF Rule Structure
Sample Rule Macro
Composing a Rule using the Rule Clerk
NOTE: The Applixware Mail Inbox is not supported in the NT or Windows 95 versions of Applixware.

Autodisposition Examples

Here are some examples of using autodisposition rules:

· If you are out of the office for a period of time, you can forward some or all mail to someone else.

· You can import all mail sent from your supervisor into a mail folder named Immediate. However, everything received from John Doe is moved directly into the Wastebasket.

· All messages having an attached spreadsheet are forwarded to Peter Main (unless, of course, the message was sent by Peter Main).

· All messages received while you are on vacation are placed in a Vacation folder and the senders of the messages are automatically sent a reply saying that you are on vacation and that if matters are important, to contact Helen Bates. Your vacation mail is placed in a shared folder so that Helen Bates can read it.

· On Friday, your employees send you spreadsheets that describe sales in their regions. An Autodisposition Rule consolidates these spreadsheets after they are received, graphs the result, and faxes it to your hotel.

· An employee requests a copy of a document. ELF locates the document and sends it to the requester. The request is then placed in a fulfilled folder.

For more information, select one of the following:
Autodisposition
How Autodisposition Works
ELF Rule Structure
Sample Rule Macro
Composing a Rule using the Rule Clerk

How Autodisposition Works

When a mail message is received, Mail looks at the *rules* you have defined which are currently en-abled. These rules are applied, in order of appearance in the Mail Rules dialog box, to all arriving messages. Any messages which meet none of the criteria of all enabled rules are filed in the folder

If the mail message meets the criteria you have established, the action you defined within the rule is performed. Otherwise, the message is filed in your Mail folder, or in whatever folder you specify for all unresolved messages in the **Mail Rules** dialog box.

When each enabled Rule executes, it must conclude with one of the following return statements:

· return(TRUE_TERM)

The rule conditions have been met and the proscribed action taken; do not apply anymore rules to the message.

· return(TRUE)

The rule conditions have been met and the proscribed action taken; apply the next enabled rule to the message.

· return(FALSE)

The rule conditions have not been met; apply the next enabled rule to the message.

· return(FALSE_TERM)

The rule conditions have not been met, do not apply the next enabled rule to the message. The message will be filed in your default mail folder or in whatever folder you specify for all unresolved messages in the Mail Rules dialog box.

You can change the priority of rules by changing the order of the rules within the list, enable, disable, add, delete, or edit rules. All changes are affected as soon as you execute them.

For more information, select one of the following:

Autodisposition
Autodisposition Examples
ELF Rule Structure
Sample Rule Macro
Composing a Rule using the Rule Clerk

ELF Rule Structure

The following shows an example of a simple rule structure:

```
' Include the mail header file which contains the value and format
' definitions for mail macros:
include "mail_.am"

' Name the macro
macro a_rule_name(format inb_item message)

        var sender

        ' Look at the name of the sender:
        '
        if TRIM@(message.lfrom) <> ""
                sender = message.lfrom
```

```
else
        sender = message.from

' Disregard yourself (i.e. if the sender is NOT equal to you):
'
if STRING_INDEX@(sender,"Your Username") = 0
{
        '
        ' Look at an item in the inb_item structure or look at
        ' some part of the message body and specify your criteria
        ' for messages to be acted upon.

        define_criteria@(message)

        ' process_message is a function that you write that
        ' manipulates the mail
        '
        process_message(message)

        'The mail message is then written to a folder you
        'specify, in this case, "Junk":
        '
        ADD_TO_FOLDER@(message, "Junk", TRUE)

        'If the conditions defined have been met and the proscribed
        'action taken, pass the message along to the next
        'enabled rule:
        return (TRUE)
}

' If the rule conditions were not met, pass the message along to the
' next enabled rule
return (FALSE)
```

For more information, select one of the following:

Sample Mail Rule Macros

The following macro checks incoming messages for an "urgent" flag or the word "urgent" in the subject, places the message in a folder named "Urgent." If the folder does not exist, it is created.
include "mail_.am"

macro urgent_rule(format inb_item message)

     var subj

'Convert the subject to uppercase:

     subj = UPPERCASE@(message.subject)

' If the message has the Urgent flag turned on or has the word "Urgent" ' in the Subject (making sure to type the Subject text in all uppercase):

     IF IS_URGENT@(message) OR
          STRING_INDEX@(subj,"URGENT") <> 0

' then add the message to a folder named "Urgent", and create it if it
' does not exist:

     {
          ADD_TO_FOLDER@(message,"Urgent",true)

' if the above is TRUE, do not apply any more rules to this message:
          RETURN(TRUE_TERM)
     }
' if the rule cannot be applied to this message, do nothing to the
' message except pass it along to the next enabled rule:
     ELSE
          RETURN(FALSE)

endmacro

The following macro looks for messages sent to "department", a distribution list defined outside of Applixware Mail, from a sender named "Peter Main"and puts the messages into a folder named "Junk." Further, the writer of the rule turned off all notification for the folder Junk in Folder ® **Folder Attributes**.
include "mail_.am"

```
macro JunkMail(format inb_item message)
' Define variables:
VAR sender, sender2, recipients
sender = message.lfrom
sender2 = message.from
recipients = MAIL_GET_MESSAGE_RECIPIENTS@(message)
' Look at each recipient in the array of recipients, beginning with
' recipient0, and looping through to the last recipient:
FOR index=0 to ARRAY_SIZE@(recipients)-1
' For each recipient, if the sender's long name or address is "Peter Main" OR the short name is
"pmain@bup" AND
IF (STRING_INDEX@(sender, "Peter Main")  or
    STRING_INDEX@(sender2, "pmain@bup"))  and
' the recipient is "department", then
    (STRING_INDEX@(recipients[index], "department"))
' add the message to a folder named "Junk", and create it if it
' does not already exist.
      {
            ADD_TO_FOLDER@(message, "Junk", TRUE)
' If the condition is met and the message is added to Junk, do
' not pass the message to the next enabled rule.
            return (TRUE_TERM)
      }
' Go through the loop again with each recipient, then
next index
' if the condition were not met, pass the message to the next enabled
' rule
      return (FALSE)
```

For more information, select one of the following:

Autodisposition
Autodisposition Examples
How Autodisposition Works
ELF Rule Structure
Composing a Rule using the Rule Clerk

CALS Group 4 Fax Files

Applixware supports CALS Group 4 Fax files that contain width and height information. When converting files in this format to an Applixware Graphics file, you should run the following macro:
filter_gp4_to_gr@(infile, outfile, width, height)
This macro returns the process id of the conversion, which runs in the background.

You should, however, be aware that incorrect values of width and height may cause the converter to fail.
For more information, see **FILTER_GP4_TO_GR@**.

```
/*
* the format for shared folder, infomation of shared folder
*/

format sync_table_
        folder_name,
        mod_time,
        format inb_item folder_contents,
        flag,
        access_mode,
        remote_host


/*
 * The format for an individual mail message in the Inbox
*/

format inb_item
        from,                           ' sender (for reply)
        reply,                          ' Reply by string
        date,                           ' date (text)
        subject,
        status,                         ' status flags
        file,                           ' path to attachments
        forward,                        ' forward history
        comment,
        ldate,                          ' send date as a long
        altreply,                       ' alternate recipient for reply; (corresponds to the Reply-To field)
        lfrom,                          ' long name of sender
        mid,                            ' unique message id
        response,                       ' date response sent to certmail sender
        color,                          ' inbox message display color
        reserved1,
        reserved2,
        reserved3


format folder_info_
        folder_name,
```

```
        folder_type,                        ' "private" or "shared"
        format access_data access_info,
        owner,                              ' the true folder owner
        supress,                            ' supress notifier for this folder
        unused1,
        unused2,
        unused3

format access_data
        readers,                            ' list of folder readers
        writers,                            ' list of folder writers
        deleters                            ' list of folder deleters


format rule_info_
        title,
        macro_t,
        active,
        unused1,
        unused2,
        unused3

/*
 * structure of foreign file info
 */
format ff_info_
        doctype,
        small_bitmap,
        large_bitmap,
        launch_cmd,
        launch_mode,
        unused_1,
        unused_2


/*
 * doc recipient preferences format
 */

        ' note: user's may provide an alternate function that maintains
        ' this data outside of Applixware, but it must be returned in this
        ' format...

format mail_recip_prefs_
```

```
        username,                        ' recipient name
        wp_pref,                         ' wp doc format pref
        gr_pref,                         ' gr doc format pref
        ss_pref,                         ' ss doc format pref
        wp_pref_func,                    ' wp conversion func
        gr_pref_func,                    ' gr conversion func
        ss_pref_func,                    ' ss conversion func
        mac_user,                        ' true or false (for mac hdr)
        fax_phone,                       ' phone number as dialed
        fax_longname,                    ' full name for header page
        fax_send_time,                   ' deferred send time or 0
        mime,                            ' true or false (wants mime)
        dd_pref,                         ' dd doc format pref
        dd_pref_func,                    ' dd conversion func
        unused1,
        unused2,
        unused3,
        unused4

/*
 * FAX format data - a subset of mail_recip_prefs_ passed to
 * customer supplied FAX macros
 */
format fax_info_
        recipient,                       ' recipient name
        phone,                           ' fax phone number
        longname,                        ' full name for header page
        sendtime                         ' deferred send time or 0 (now)


/*
 * the format for a selection in an editbox
 */

format selection_info_
        start_line,
        start_col,
        end_line,
        end_col



format sm_message_
        to_recips,
```

```
        cc_recips,
        bc_recips,
        files,
        ftypes,
        encodes,
        wp_pref,
        gr_pref,
        ss_pref,
        dd_pref

format mail_recip_info_
        name,
        class,
        wp_pref,
        gr_pref,
        ss_pref,
        dd_pref




/*
 * attribute definitions for mail macros
 */

define MAIL#SUBJECT_              0
define MAIL#SENDER_               1
define MAIL#DATE_                 2
define MAIL#COMMENT_              3
define MAIL#BODY_                 4

define MAIL#ALL_DATES_            0
define MAIL#YESTERDAY_            1
define MAIL#TODAY_                2
define MAIL#DATE_RANGE_           3

define TRUE_TERM                  2
define FALSE_TERM                 3
define OVERRIDE_SAFETYNET         10    ' allows for disposal of message
define SUPRESS_NOTIFIER           100   ' no notifier for this message


define LAUNCH_TYPE_SHCMD          0     ' launch foreign file w/shell cmd
define LAUNCH_TYPE_AXMACRO        1     ' launch foreign file as * macro
```

```
/*
 * Shared folder data...
 */

define BB#SVR 4137

define BB#SERVER_NAME NXLT("bb server")

' request types...
define BB#REGISTER              0
define BB#SYNC                  1
define BB#GET_INFO             2
define BB#TERM                  3
define BB#FOLDER_LIST          4
define BB#MESSAGE_LIST         5
define BB#MESSAGE_INFO         6
define BB#FOLDER_INFO          7
define BB#MESSAGE_UPDATE       8
define BB#MESSAGE_ADD          9
define BB#MESSAGE_DELETE       10
define BB#FOLDER_ACCESS        11
define BB#ENV_INFO             12
define BB#GET_FILE             13
define BB#DELETE_FILE          14
define BB#GET_VERSION          15

' ack types...
define BB#SYNC_ACK             21
define BB#INFO_ACK             22
define BB#TERM_ACK             23
define BB#UPDATE_ACK           24
define BB#UPDATE_ERROR         25
define BB#UNKNOWN_CMD          26


format bb_request_
        caller,
        type,
        message,
        unused1,
        unused2,
        unused3
```

```
format pending_message_
        to_recips,
        cc_recips,
        bc_recips,
        subject,
        msgbody,
        attfiles,
        reply_state,
        reply_text,
        urgent_state,
        certified_state,
        encrypt_state,
        outbox_state,
        addrecip_state,
        is_signed

/*
 * format for mailed document conversion options
 */
        format mail_format_array_
            name,               ' name of conversion format (text)
            filtmac,            ' name of conversion function
            extension,          ' extension for converted document
            doctype             ' Applixware magic number for output file
```

READ_SELECTED_ATTACHMENTS@

Reads the selected attachment

Format   READ_SELECTED_ATTACHMENTS@( )

REGEN_HOOK_INFO@

Clears the hooks.dat cache

Format   REGEN_HOOK_INFO@()

Description Clears the hooks.dat cache. hooks.dat allows you to automatically import non-Applixware attached files into an Applixware application or launch the document's native application directly from the Inbox. If a change is made in the hooks.dat file, use this macro to clear the cache and force the new information to be generated when a foreign attachment is displayed in the Inbox.

Previous macros

The following is a listing of macros available in previous versions of Applixware, and their Applix*ware* 3.2 equivalents.

Macro	Applixware 3.2 Equivalent Macro
ADD_TO_FOLDER@	**MAIL_ADD_TO_FOLDER@**
DISPLAY_ALL@	**MAIL_DISPLAY_ALL@**
GET_FOLDER_ATTRS@	**MAIL_GET_FOLDER_ATTRS@**
GET_MSG_LOG@	**MAIL_GET_MSG_LOG@**
HAS_ATTACHMENTS@	**MAIL_HAS_ATTACHMENTS@**
INB_READ_UMAIL@	**INB_READ_UMAIL@**
INB_RECORD_MACRO@	**INBOX_RECORD_MACRO@**
IS_CERTIFIED@	**MAIL_IS_CERTIFIED@**
IS_DOCUMENT@	**MAIL_IS_DOCUMENT@**
IS_ENCRYPTED@	**MAIL_IS_ENCRYPTED@**
IS_SIGNED@	**MAIL_IS_SIGNED@**
IS_TAGGED@	**MAIL_IS_TAGGED@**
IS_URGENT@	**MAIL_IS_URGENT@**
LIST_OF_FOLDERS@	**MAIL_LIST_OF_FOLDERS@**
LIST_OF_STUFF@	**MAIL_LIST_OF_STUFF@**
RECOVER@	**MAIL_RECOVER@**
REHASH_GLOBAL_USERLIST@	**MAIL_REHASH_GLOBAL_USERLIST@**
SIZE_OF_FOLDER@	**MAIL_SIZE_OF_FOLDER@**
UPDATE_MSG_LOG@	**MAIL_UPDATE_MSG_LOG@**
WRITE_FOLDER_ATTRS@	**MAIL_WRITE_FOLDER_ATTRS@**