

---

## Workgroup Extensions for Microsoft Excel

The Workgroup Extensions let Microsoft Excel developers include electronic mail and simple network management functionality in their custom applications. With Microsoft Excel macros, you can now:

- Send messages, including attached files, to multiple users on your network.
- Search for messages stored in you inbox.
- Connect to, display, and disconnect from network resources.
- Copy files between computers running on a network.

To use the Workgroup Extensions, you must have:

- Requirements for Message Functions: A MAPI-compliant electronic mail program, such as the mail client of Microsoft Windows for Workgroups 3.1 or Microsoft Mail for Windows 3.0.
- Requirements for Network Functions: Microsoft Windows for Workgroups 3.1 or networks supported by Microsoft Windows 3.1.

Before using the Workgroup Extensions you should be familiar with the concepts and use of the Microsoft Mail client application and understand how to connect and disconnect Lanman network resources.

For information on the complete MAPI specification and more detailed conceptual information on messaging, refer to the *Microsoft Mail Technical Reference*.

### Functions Included in the Workgroup Extensions

This section lists, in alphabetical order, the message and network functions included in the Workgroup Extensions:

Message Function	Description
------------------	-------------

GET.MAIL	Returns information about the Workgroup Extensions
GET.MAIL.ATTACHMENT	Returns information about the attachment(s) to the current message
GET.MAIL.MESSAGE	Returns information about the current message
GET.MAIL.RECIPIENT	Returns information about the recipients of the current message
MAIL.ADDRESS	Displays the mail addressing dialog box
MAIL.DELETE.MESSAGE	Deletes the current message and replaces the current message with a new, blank message
MAIL.DETAILS	Displays information about a specified recipient
MAIL.FIND.NEXT	Finds messages stored in the inbox
MAIL.LOGOFF	Cancels the current mail session
MAIL.LOGON	Establishes a mail session for Microsoft Excel
MAIL.NEW.MESSAGE	Clears the current message and replaces it with a new,

---

	blank message
MAIL.RESOLVE.RECIPIENT	Resolves the address of the recipients of the current message
MAIL.SAVE.MESSAGE	Saves the current message to the mailbox
MAIL.SEND.MESSAGE	Sends the current message
SET.MAIL.ATTACHMENT	Attaches documents to the current message
SET.MAIL.MESSAGE	Sets fields of the current message
SET.MAIL.RECIPIENT	Sets information about the recipient(s) of the current message

**Network Function****Description**

---

FILE.COPY	Copies a file
GET.NET	Returns information about the network
GET.TEMP.FILE	Creates a temporary file

NET.BROWSE	Displays dialog boxes that let the user change network connections and find network resources.
NET.CONNECT	Connects a computer to a network resource
NET.DISCONNECT	Disconnects a computer from a network resource

## Loading the Workgroup Extensions

To use the Microsoft Excel Workgroup Extensions, first copy the file XLMAPI.XLL from the Workgroup Extensions diskette onto your hard drive. If you want the Workgroup functions loaded automatically whenever you start Excel, place the XLMAPI.XLL file in your \XLSTART sub-directory. Otherwise, the Workgroup Extensions can be loaded into Excel by using the File-Open command or Excel's Add-in Manager. To load the Workgroup Extensions from the macro language use the REGISTER function:

```
REGISTER("C:\EXCEL\XLMAPI.XLL")
```

## Sending Messages Using the Workgroup Extensions

There are seven steps to creating and sending messages using the Workgroup Extensions with your Microsoft Excel macros.

1. Start a mail session for Microsoft Excel with the MAIL.LOGON function.
2. Create a new current message with the MAIL.NEW.MESSAGE function
3. Build the recipient list, contents, and attachments of your message using the SET.MAIL.MESSAGE function. You need to use this function once for each part of the message you are building.
4. If you want, you can add to or change the attachments to your message using the SET.MAIL.ATTACHMENT function.

5. If you want, you can add to or change the recipients of your message using the SET.MAIL.RECIPIENT function. You may also use MAIL.ADDRESS, MAIL.DETAILS, and MAIL.RESOLVE.RECIPIENT to let the user choose recipients, see information on recipients and resolve recipient names.
6. When you are done creating your message, you can send it using the MAIL.SEND.MESSAGE function.
7. When you are done working with messages, close the mail session using the MAIL.LOGOFF function.

## Understanding Mail Sessions

The Workgroup Extensions messaging functions require that a mail session be established before they can be used. A mail session validates the identity of the sender and indicates which message store to use when finding and saving messages. Mail sessions with the Workgroup Extensions require a password for security.

The Workgroup Extensions allow only one mail session per instance of Excel. Macro Add-in using the Workgroup Extensions must share this common session by not assuming exclusive ownership. Add-ins should:

- Check whether a session is already established by another add-in. If appropriate, this session can be used. GET.MAIL(4) indicates whether a session is currently present.
- Close the existing session and create a new one if the existing session username is incorrect. GET.MAIL(5) returns the name of the current session.
- Close sessions when they are no longer needed.

## Understanding Messages

### The Current Message

The Excel Workgroup Extensions maintain a single *current message* which is maintained in memory. The current message corresponds to a note window in the Microsoft Mail client application. When you use the Workgroup Extensions messaging functions you are implicitly editing this message. Unless explicitly saved, any changes to the current message are discarded when the message is cleared.

When a mail session is established there is no current message. A blank current message is created for editing with MAIL.NEW.MESSAGE. When a message is found in the inbox with MAIL.FIND.NEXT that message is copied into memory.

In both cases, changing the current message automatically clears the previous current message.

## Recipients

Messages are sent based on a recipient list. Each recipient is identified with a *friendly name* and an address. A friendly name is the descriptive name of the user. Friendly names can be ambiguous, therefore additional information is required to guarantee unique addressing. The address is a unique string that contains the account name of the user and must be unambiguous. The address string format varies depending on the configuration of the mail transport system and must be obtained from the Workgroup Extensions before it can be used.

When setting and storing message recipients you should use both the name and the address strings to avoid ambiguous naming problems. The Workgroup Extensions return both strings for recipients that are picked using the mail address book or for recipients of messages read from the inbox. You should use both strings when passing recipients to the Workgroup Extensions, although both strings are not required. If you do not supply an address string the Workgroup Extensions will attempt to automatically resolve the name. Should the name be ambiguous this resolution process may cause the function to fail or require that the user choose between possible candidates. To avoid this interruption you can force name resolution by using the MAIL.RESOLVE.RECIPIENT function.

## Attachments

Attachments are files that are sent with messages. Attachments appear as icons within the body text and are positioned by replacing one character of the message with the attachment. The Excel Workgroup Extensions do not support OLE attachments but instead require that attachments be true DOS files.

When a message is read from the inbox its attachments are stored as temporary files. These temporary files are maintained by the Workgroup Extensions and are automatically deleted when the current message is changed. If you want to store the attachment file for future use you must copy it using the FILE.COPY function.

If you provide your own attachment files to the Workgroup Extensions, you are responsible for deleting them using the FILE.DELETE macro function. Supplied attachments should not be moved, renamed, or deleted until after the message containing the attachments is sent or saved.

## Understanding Recipient and Attachment Arrays

Recipients and attachments are commonly retrieved and set as a list. Lists of

recipients and attachments are stored as *array constants*. An array constant works much like a range of cells. For more information on array constants, see Chapter 5, “Creating a Worksheet,” in the *Microsoft Excel 4.0 User’s Guide, Book 1*.

Excel functions that are useful with arrays include COLUMNS, HLOOKUP, INDEX, LOOKUP, MATCH, ROWS, and VLOOKUP. For information on these functions, see the *Microsoft Excel 4.0 Function Reference*.

## Recipient Arrays

In the recipient array, the information for each recipient is stored in a separate row. Each row has three columns:

- The first column stores the name of recipient as a text string. This column can be left blank, but the column itself is required.
- The second column stores the recipient’s mail address as a text string. This column can be empty if there is an entry in the first column.
- The third column stores the recipient class as a number: 1 = To recipient, 2 = CC recipient, 3 = BCC (blind carbon copy) recipient. When this column is blank or missing, it is assumed to be 1.

Recipient arrays are used for both providing information to functions and returning information from functions. When a recipient array is used to provide information to a function, only the first column of the array is required. When a function returns a recipient array, it returns all three columns of the array.

Functions that set or use recipient arrays are GET.MAIL.MESSAGE, GET.MAIL.RECIPIENT, MAIL.ADDRESS, MAIL.RESOLVE.RECIPIENT, SET.MAIL.MESSAGE, SET.MAIL.RECIPIENT.

## Attachment Arrays

Attachment arrays store information describing attachments to the current message. Like recipient arrays, attachments arrays have three columns:

- The first column stores the MS-DOS pathname of the attachment as a text string. You can use relative pathnames, but fully-qualified pathnames are recommended.
- The second column stores a label for the attachment as a text string. If this column is blank, the filename of the attachment is used as the label. Use labels when attachments have temporary filenames that don’t describe the contents of the attachment. Note that it is not necessary that the label be a valid DOS file name, for example

"Russell's message" would be appropriate.

- The third column stores the character position of the attachment in the message as a number from 1 to the length of the message text. Using -1 as the position places the attachment before any body text in the message. Using -2 places the attachment after any body text at the end of the message. When the contents of this column is blank, missing, or larger than the length of the message, it is assumed to be -2. The attachment icon replaces the character of the message body text.

Functions that set or use attachment arrays are GET.MAIL.ATTACHMENT, GET.MAIL.MESSAGE, SET.MAIL.ATTACHMENT, SET.MAIL.MESSAGE.

## Handling Message Text

Excel only permits strings that have a maximum of 255 characters. If you wish to build message bodies that are greater than 255 characters in length, you must do so by manipulating text files with the FOPEN, FCLOSE, FREAD, and FWRITE macro functions.

## Workgroup Extensions Functions

---

### FILE.COPY

Macro Sheets Only

Copies a file to a new name and/or destination. Use FILE.COPY as an equivalent to an operating system file copy command, especially to share copies of files with other users across a network.

**FILE.COPY**(*source\_file\_text*, *dest\_file\_text*, *overwrite\_dest\_logical*)

*Source\_file\_text* is the name of the original file you want to copy.

*Dest\_file\_text* is the name and location of the copied file. If *dest\_file\_text* consists only of a directory name, a copy of the source file with the source file name is made in the destination directory. If *dest\_file\_text* consists only of a drive letter, a copy of the source file with the source file name is made in the current directory of the destination drive. If the destination directory or drive does not exist, FILE.COPY returns the #VALUE! error value.

*Overwrite\_dest\_logical* is a logical value that specifies whether you want to replace a file in the destination that has the same name as *dest\_file\_text*.



- If *overwrite\_dest\_logical* is TRUE, FILE.COPY overwrites a file in the destination that uses the same name as your destination file and returns the value TRUE.
- If *overwrite\_dest\_logical* is FALSE or omitted, and a duplicate filename exists in the destination, FILE.COPY does not overwrite the file and returns the value FALSE.

The following example makes a backup copy of the file SALES.XLS to SALES.BAK, replacing any previously existing version of SALES.BAK:

```
FILE.COPY("SALES.XLS","SALES.BAK",TRUE)
```

The following example copies SALES.XLS to the current directory of drive G, unless a file named SALES.XLS already exists in the destination:

```
FILE.COPY("SALES.XLS","G:")
```

## Function

## Description

---

### DIRECTORY

Sets the current drive and directory to the specified path and returns the name of the new directory as text

### FILES

Returns a horizontal array of files in the current directory

### GET.TEMP.FILE

Creates a uniquely named temporary file and returns the file name as text

### SAVE.AS

Saves a document and allows you to specify the name, file type, password, backup file, and location of the document

---

## GET.MAIL

Macro Sheets Only

Returns information about the Workgroup Extensions. Use GET.MAIL to get

Microsoft Excel Function Reference

---

information about the current Mail session.

**GET.MAIL(*type\_num*)**

*Type\_num* is a number that specifies what type of information you want about the Workgroup Extensions. The following list shows the possible values of *type\_num* and the corresponding results.

<i>Type_num</i>	Returns
1	Number indicating the type of error caused by the last Workgroup Extensions function: 0 = Success 1 = User aborted 2 = Failure 3 = Login failure 4 = Disk full 5 = Insufficient memory in MAPI 6 = Access denied 8 = Too many sessions 9 = Too many files 10 = Too many recipients 11 = Attachment not found 12 = Attachment open failure 13 = Attachment write failure 14 = Unknown recipient 17 = Invalid message 18 = Text too large 19 = Invalid mail session 20 = Type not supported 21 = Ambiguous recipient 23 = Network failure 25 = Invalid recipients

## Excel Workgroup Extensions 11

1000 = Memory failure in Workgroup Extensions

1001 = MAPI not installed

1002 = No mail session established

1003 = Out of range index value

1004 = Invalid argument type

1005 = File cannot be written or read

1006 = No current message present

Use error codes to provide detailed information to the user (through an error handler) and attempt recovery.

2	Version number of Workgroup Extensions as text.
3	If MAPI or a Microsoft Mail compliant transport layer is installed and available, returns TRUE; otherwise FALSE.
4	If a Mail session is currently established, returns TRUE; otherwise FALSE.
5	Returns the Friendly Name of the current Mail session as text; if there is no current Mail session, returns FALSE.
6	The number of unread messages in the Mail inbox.

If a Mail session is established, then:

GET.MAIL(4) equals TRUE

If a Mail session is established, and there are 10 unread messages in the inbox, then:

GET.MAIL(6) equals 10

Function	Description
----------	-------------

## Microsoft Excel Function Reference

---

GET.MAIL.ATTACHMENT Returns information about an attachment in the current Mail message

GET.MAIL.MESSAGE Returns information about the current Mail message

GET.MAIL.RECIPIENT Returns information about the recipients of the current Mail message

MAIL.LOGON Establishes a mail session

---

---

## GET.MAIL.ATTACHMENT

Macro Sheets Only

Returns information about an attachment in the current Mail message. Use GET.MAIL.ATTACHMENT to handle attachments within Mail messages.

### GET.MAIL.ATTACHMENT(*type\_num*, *index\_num*)

*Type\_num* is a number specifying the type of information you want about an attachment in the current Mail message. The following list shows the *type\_num* values and their corresponding results:

<i>Type_num</i>	Returns
1	Returns the file and path name of the attachment as text.
2	Returns the attachment label as text.
3	The character position of the attachment within the message body as an integer from 1 to GET.MAIL.MESSAGE(16). This value is -1 if the attachment is at the start of the message, and -2 if the attachment is at the end.
4	If the attachment file will be automatically deleted with the message, returns TRUE; if the attachment file must be deleted by the user, returns FALSE.

*Index\_num* is an integer that specifies which attachment you want information about in the current Mail message. You can use GET.MAIL.MESSAGE(10) to return the total number of attachments in a message and GET.MAIL.MESSAGE(8) to return a list of all attachments.

If the attachment label of the first attachment in the current Mail message is "Budget Template," then:

Microsoft Excel Function Reference

---

GET.MAIL.ATTACHMENT(2,1) equals "Budget Template"

Function

Description

---

GET.MAIL

Returns information about the current Mail session

GET.MAIL.MESSAGE

Returns information about the current Mail message

GET.MAIL.RECIPIENT

Returns information about the recipients of the current Mail message

---

GET.MAIL.MESSAGE

Macro Sheets Only

Returns information about the current Mail message. This function can return recipient and attachment arrays. For information on these, see “Understanding Recipient and Attachment Arrays” earlier in this document.

**GET.MAIL.MESSAGE(*type\_num*)**

*Type\_num* is a number specifying the type of information you want about the current Mail message. The following list shows the *type\_num* values and their corresponding results:

*Type\_num*

**Returns**

---

1

Returns the message subject as text.

- 
- |   |  |
|---|--|
| 2 | Returns the first 255 characters of the message as text. Use <i>type_num</i> 3 to access the full message text.  |
| 3 | Creates a temporary file containing the full text of the current message and returns the file name. Returns #N/A if the message body contains no text. You can use Microsoft Excel file functions to access message text in the temporary file. This file is maintained by the Workgroup Extensions and is deleted when the current message is changed.  |
| 4 | Returns the message type, such as "IPM", as text. The message type is used by the mail transport to support custom message forms and handling of messages. Message types begin with either "IPM." or "IPC." and are optionally followed by application-specific information such as "IPM.Microsoft Excel.Document Routing". IPM messages appear in the inbox like messages sent from a Mail client. IPC messages do not appear in the inbox and are useful for interprocess communication. Messages sent from the Mail client have the type "IPM.Microsoft Mail.Note". Refer to the <i>Microsoft Mail Technical Reference</i> for more information on message types. |
| 5 | Message conversation ID as text. Returns the empty string if there is no conversation ID.  |
| 6 | If the message is marked Return Receipt, returns TRUE, else FALSE.   |
| 7 | Returns a recipient array containing all recipients of the current message. If there are no recipients, #N/A is returned. Use MAIL.GET.RECIPIENT to obtain information about a single recipient of the current message.  |
| 8 | Returns an attachment array containing all attachments of the current message. If there are no attachments, returns #N/A. Use MAIL.GET.ATTACHMENT to obtain information about a single attachment.   |
| 9 | Number of recipients of the current message.   |

## Microsoft Excel Function Reference

---

- |    |   |
|----|---|
| 10 | Number of attachments in the current message.   |
| 11 | If the current message has been sent and received at least once, returns TRUE; otherwise FALSE. Useful for determining whether the message is a sent or composed but unsent message. If a sent message has been edited, still returns TRUE.   |
| 12 | Returns the date the current message was received or saved as a date serial number. Returns #N/A if the message has not been sent or saved.   |
| 13 | Returns the name of the sender as text. Returns #N/A if the message has not been sent or saved.   |
| 14 | Returns the address of the sender as text, or a null string if the name suffices. Returns #N/A if the message has not been sent or saved.   |
| 15 | If the message is unread, returns TRUE; otherwise FALSE.  |
| 16 | Returns the number of text characters in the message body as an integer.  |
| 17 | If the message is present in the inbox, returns TRUE; if the message is dynamically created and is not present in the inbox, returns FALSE.   |
| 18 | Creates a temporary file containing the full text of the current message and returns the file name. This function differs from function 3 in that place holder text for attachments is inserted within the body text. The resulting body text is similar to the attachment place holders inserted by Mail clients when replying to a message with attachments. Returns #N/A if the message body contains no text and no attachments. You can use Microsoft Excel file functions to access message text in |



## Excel Workgroup Extensions 17

the temporary file. This file is maintained by the Workgroup Extensions and is deleted when the current message is changed.

If the current message is unsent and contains two attachment files, one at the beginning and one at the end:

```
GET.MAIL.MESSAGE(8) equals
    {"C:\TMP\BUDGET.XLS","Departmental Budget",-1;
     "C:\TMP\COMMENTS.DOC","Comments",-2}
```

```
GET.MAIL.MESSAGE(10) equals 2
```

```
GET.MAIL.MESSAGE(11) equals FALSE
```

## Function

## Description

---

GET.MAIL

Returns information about the current Mail session

GET.MAIL.ATTACH  
MENT

Returns information about an attachment in the current Mail message

GET.MAIL.RECIPIEN  
T

Returns information about the recipients of the current Mail message

---

GET.MAIL.RECIPIENT

Macro Sheets Only

Returns information about recipients of the current Mail message. Use GET.MAIL.RECIPIENT to get detailed information about a particular message recipient.

**GET.MAIL.RECIPIENT(*type\_num*, *index\_num*)**

## Microsoft Excel Function Reference

*Type\_num* is a number that specifies what information you want about a recipient of the current Mail message. The following list shows the possible values of *type\_num* and their corresponding results.

<i>Type_num</i>	Returns
1	Returns the name of a recipient as text.
2	Returns the address of the recipient as text.
3	Returns the type of the recipient as a number: 1 = To 2 = Cc (carbon copy) 3 = Bcc (blind carbon copy)

*Index\_num* is the number of the recipient you want information about. You can use GET.MAIL.MESSAGE(9) to return the total number of recipients and GET.MAIL.MESSAGE(7) to return a list of recipient names.

If the current message is addressed to Dennis Smith and Ellen Kim on the To line and Dave Martinez on the Cc line:

GET.MAIL.RECIPIENT(1,2) equals "Ellen Kim"

GET.MAIL.RECIPIENT(3,3) equals 2

Function	Description
GET.MAIL	Returns information about the current Mail session
GET.MAIL.ATTACHMENT	Returns information about an attachment in the current Mail message

GET.MAIL.MESSAGE Returns information about the current Mail message

---

---

## GET.NET

Macro Sheets Only

Returns information about the network. Use GET.NET to determine connections to network resources or to detect macro errors caused by network functions available through the Workgroup Extensions.

**GET.NET**(*type\_num*, *name\_text*)

*Type\_num* is a number that specifies what type of information you want about the network. The following list shows the possible values of *type\_num* and the corresponding results:

<i>Type_num</i>	Returns
1	A number indicating the last macro error caused by a network function available through the Workgroup Extensions: 0 = Success 1 = Not supported 2 = Network error 5 = Bad value 6 = Bad password 7 = Access denied 11 = Out of memory 12 = User canceled 48 = Local device not connected 49 = Open files are present on network drive 50 = Bad network name 51 = Bad local device name 52 = Local device already connected 1000 = Memory failure in Workgroup Extensions 1004 = Invalid argument type
2	If the network is compatible with the NET.CONNECT and NET.DISCONNECT functions, returns TRUE; otherwise FALSE.
3	Returns the name of the network resource connected to the drive <i>name_text</i> as text in the form "\\server\share".

- 4 Returns the name of the first local device connected to the network resource *name\_text* as text (in order from A: to Z: or LPT1: to LPT9:). The resource *name\_text* must be in the form "\\server\share".
- 5 Returns a two-column array of network device connections currently present. The first column contains the local device name as text, and the second column contains the network resource name as text. Returns #N/A if no network resources are in use.

*Name\_text* is text for a local device name or network resource name, depending on which *type\_num* argument you use. Local device names take the form such as "G:" or "LPT1:" and network resource names take the form "\\server\share".

If the local drive G: is assigned to the server \\FINANCE\PUBLIC, and LPT1: is assigned to \\FINANCE\LASERJET then:

GET.NET(3,"G:") equals "\\FINANCE\PUBLIC"

GET.NET(4,"\\FINANCE\PUBLIC") equals "G:"

GET.NET(5) equals  
{ "G:","\\FINANCE\PUBLIC"; "LPT1:","\\FINANCE\LASERJET" }

Function	Description
NET.BROWSE	Displays dialog boxes for browsing, connecting to, or disconnecting from networks
NET.CONNECT	Connects to a network resource
NET.DISCONNECT	Disconnects from a network resource

Microsoft Excel Function Reference

---

**GET.TEMP.FILE**

Macro Sheets Only

Creates and returns a unique temporary filename that does not currently exist. The temporary file is stored in the specified directory used by Windows for temporary files. Use GET.TEMP.FILE to manage long text for a Mail message.

**GET.TEMP.FILE()**

---

It's up to you to delete the temporary file when you no longer need it, using the FILE.DELETE function.

---

To create a temporary file, use the following function:

GET.TEMP.FILE()

**Function****Description**

---

FILE.COPY

Copies a file to a new destination.

FILE.DELETE

Deletes a file from disk.

FILES

Returns a horizontal array of files in the specified directory.

GET.MAIL.MESSAGE

Returns information about the current Mail message

---

**MAIL.ADDRESS**

Macro Sheets Only

Displays the Mail addressing dialog box. Use MAIL.ADDRESS to allow the user to construct or modify the recipients of the current message.

**MAIL.ADDRESS**(*recips\_array*, *caption\_text*, *class\_fields\_num*)

*Recips\_array* is an array of text names for recipients, used to initialize recipients in the addressing dialog box. If *recips\_array* is omitted, the addressing dialog box initially displays no recipients. For information on recipient arrays, see “Understanding Recipient and Attachment Arrays” earlier in this document.

*Caption\_text* is a text value specifying the desired title of the addressing dialog box. If *caption\_text* is omitted or is a null string, the dialog box title defaults to "Address Book".

*Class\_fields\_num* is a number specifying the default type of addressing dialog box displayed. *Class\_fields\_num* is forced to the maximum recipient class number in the incoming recipient array. If *class\_fields\_num* is omitted, it is assumed to be 1. The following list shows the possible values of *class\_fields\_num* and their corresponding results.

*Class\_fields\_num*      **Returns**

---

0	Addressing dialog box allows only browsing. For <i>class_fields_num</i> of 0, <i>recips_array</i> is not used and #N/A is returned.
1	Displays the To box of the addressing dialog box.
2	Displays the To and Cc boxes of the addressing dialog box.
3	Displays the To, Cc, and Bcc boxes of the addressing dialog box.
4	Displays edit controls for all types of addressing supported by your electronic mail system.

If the user chooses the OK button in the addressing dialog box, MAIL.ADDRESS

## Microsoft Excel Function Reference

---

returns a recipient array of the selected addresses, or #N/A if no addresses were selected. If the user chooses the Cancel button, MAIL.ADDRESS returns FALSE.

To display the Mail addressing dialog box with no special settings, use the following function

```
MAIL.ADDRESS()
```

To display the Mail addressing dialog with the Finance Department as recipients and a special title, use the following function

```
MAIL.ADDRESS(FinanceArray, "Finance Dept", 4)
```

### Function

### Description

---

MAIL.DETAILS

Displays the Mail details dialog box for a recipient

MAIL.RESOLVE.RECIPIENT

Resolves the mail addresses specified in recipient array

---

## MAIL.DELETE.MESSAGE

Macro Sheets Only

Deletes the current message using MAIL.FIND.NEXT, or if the message has been saved, using MAIL.SAVE.MESSAGE. If the current message was obtained from the Mail inbox, MAIL.DELETE.MESSAGE deletes the saved message from the inbox.

### MAIL.DELETE.MESSAGE()

When the current message is deleted, any temporary attachment and message body text files set up by macro functions available through the Workgroup Extensions are also deleted. Attachment files provided by user are not deleted. If the current message is not in the inbox, MAIL.DELETE.MESSAGE has the same effect as MAIL.NEW.MESSAGE.



To delete the current message, use the following function:

MAIL.DELETE.MESSAGE()

Function

Description

---

MAIL.FIND.NEXT

Accesses messages stored in the inbox

MAIL.NEW.MESSAGE

Replaces the current message with a new, blank message

MAIL.SAVE.MESSAG  
E

Saves the current message in the inbox

---

## MAIL.DETAILS

Macro Sheets Only

Displays the details about a Mail recipient in a dialog box.

**MAIL.DETAILS**(*name\_text*, *address\_text*)

*Name\_text* is the name of the recipient you want details about, as text.

*Address\_text* is the address text, if needed, of the recipient you want details about, as text.

You must specify either the *name\_text* or *address\_text* argument.

If the recipient is successfully resolved, MAIL.DETAILS displays the Mail details dialog box and returns TRUE. If the recipient name is ambiguous and error dialog boxes are enabled, the resolve dialog box is displayed to allow the user to choose the correct recipient name. If the user chooses the Cancel button, MAIL.DETAILS returns FALSE.

The following example displays the Mail details dialog box for information about Dennis Smith:

Microsoft Excel Function Reference

---

MAIL.DETAILS("Dennis Smith")

Function	Description
----------	-------------

MAIL.ADDRESS	Displays the Mail addressing dialog box
--------------	---

MAIL.RESOLVE.RECIP IENT	Resolves the recipients specified in an input array
----------------------------	---

---

### MAIL.FIND.NEXT

Macro Sheets Only

Accesses messages stored in the Mail inbox, according to the search conditions you specify. Use MAIL.FIND.NEXT to search the inbox for messages. MAIL.FIND.NEXT sets the current message to the result of the search.

**MAIL.FIND.NEXT**(*first\_message\_logical*, *sender\_search\_text*, *to\_search\_text*, *cc\_search\_text*, *subject\_search\_text*, *file\_type\_search\_text*, *message\_type\_text*, *unread\_logical*, *ordered\_logical*, *mark\_read\_logical*)

*First\_message\_logical* controls searching for messages matching the specified criteria (see explanatory text in the Remarks section).

*Sender\_search\_text* is the name of the sender to search for, as text. If omitted, all are matches.

*To\_search\_text* is the name of a recipient on the To line to search for, as text. If omitted, all messages are matches.

*Cc\_search\_text* is the name of a recipient on the Cc line to search for, as text. If omitted, all messages are matches.

*Subject\_search\_text* is the message subject to search for, as text. If omitted, all messages are matches. If you're searching for replied or forwarded messages by subject, you need to consider those message subject prefixes, such as "RE:"

*File\_type\_search\_text* is a list of file extension types of attachments to search

for, as text of the form "XLS,XLM" for example. In order to match, the message must contain at least one attachment with the same file extension as at least one of the extensions provided by `file_type_search_text`. If omitted, all messages are matches.

`Message_type_text` is the message type to search for, as text of the form "IPM" for example. If omitted, all messages are matches.

`Unread_logical` is a logical value specifying whether to search for read or unread messages.

- If `unread_logical` is TRUE, only unread messages are matches.
- If `unread_logical` is FALSE or omitted, all messages are matches.

`Ordered_logical` is a logical value specifying the order in which messages are matched.

- If `ordered_logical` is TRUE, messages are matched in the order they were received .
- If `ordered_logical` is FALSE or omitted, no order is guaranteed.

`Mark_read_logical` is a logical value specifying whether found messages are marked as read.

- If `mark_read_logical` is TRUE, messages are always marked as read when they are found.
- If `mark_read_logical` is FALSE or omitted, matched messages are not marked as read. If a found message is already marked as read, no change occurs.

This function returns TRUE if a matching message is found, FALSE if no more matching messages are available, or #VALUE! if an error occurs.

To start searching for messages, specify TRUE for `first_message_logical`. When `first_message_logical` is TRUE, MAIL.FIND.NEXT returns TRUE when it finds the first matching message. The first matching message then becomes the current message. Once the first matching message is found, subsequent messages matching the criteria can be found by setting `first_message_logical` to FALSE. When `first_message_logical` is FALSE, MAIL.FIND.NEXT returns TRUE for the next matching message and makes that message the current message. When there are no more matching messages, returns FALSE.

When MAIL.FIND.NEXT returns FALSE or #VALUE!, the current message is unmodified. If TRUE is returned, the matching message becomes the current message. Any previous current message is freed before the new message is read. When the current message is deleted, any temporary attachment and message body text files set up by macro functions available through the Workgroup Extensions are also deleted. Attachment files provided by user are not deleted.

## Microsoft Excel Function Reference

---

This function is thus suitable for using in a WHILE() loop.

Use of the search parameters to filter messages is significantly faster than reading all messages and then manually comparing the subject or sender by using GET.MAIL.MESSAGE.

To display the subjects of all messages in the order they were received, where the messages have an Excel worksheet attachment:

```
Done=MAIL.FIND.NEXT(TRUE,,,,"XLS" ,,TRUE)
WHILE(Done=TRUE)
    ALERT("Subject:"&GET.MAIL.MESSAGE(1))
    Done=MAIL.FIND.NEXT(FALSE,,,,"XLS" ,,TRUE)
NEXT()
```

Function	Description
<hr/>	
GET.MAIL.MESSAGE	Returns information about the current message
MAIL.SAVE.MESSAG E	Saves a message in the Mail inbox

---

---

**MAIL.LOGOFF**

Macro Sheets Only

Ends the established Mail session.

**MAIL.LOGOFF()**

If the Mail session is successfully ended, MAIL.LOGOFF returns TRUE. When the session is ended, the current message is deleted, and any temporary attachment and message body text files set up by macro functions available through the Workgroup Extensions are also deleted. Attachment files provided by user are not deleted.

**Function****Description**

---

**MAIL.LOGON**

Starts a Mail session

---

**MAIL.LOGON**

Macro Sheets Only

Establishes a mail session for the current instance of Microsoft Excel.

**MAIL.LOGON**(*name\_text*, *password\_text*, *download\_logical*)**MAIL.LOGON?**(*name\_text*, *password\_text*, *download\_logical*)

*Name\_text* specifies the username of the mail account (not the friendly name). If *name\_text* is omitted, displays the logon dialog.

*Password\_text* gives the password for the account and may be omitted if the user is to enter the password. This parameter is ignored when the question mark form is used.

*Download\_logical* forces any new mail to be downloaded to the user's mailbox. Setting this value to TRUE causes new mail to be downloaded at the cost of some extra processing time. Using FALSE or omitting the parameter does not guarantee that new mail is downloaded.

Microsoft Excel Function Reference

---

Returns TRUE on success, FALSE if the user cancels the logon dialog box, or #VALUE! upon error. If the function fails when the regular form of the function is used and error trapping is turned off, the function returns #VALUE!.

Once established, a mail session remains until either MAIL.LOGOFF is called or the Workgroup Extensions or Microsoft Excel is closed.

If a session already exists when this function is called, the existing session is silently closed *before* attempting to open the new session.

To start a new mail session, be prompted for your password, and download your new mail, use the following example:

```
MAIL.LOGON(,TRUE)
```

For Heikki Bardell to start a new mail session and download her new messages, she would use the following function:

```
MAIL.LOGON("HeikkiBa",,TRUE)
```

Function	Description
MAIL.LOGOFF	Closes a Mail session

---

**MAIL.NEW.MESSAGE**

Macro Sheets Only

Clears the current message and replaces it with a new, blank message. Use this function to simply clear out the previous message and to guarantee a clean slate when constructing a new message.

**MAIL.NEW.MESSAGE()**

Returns TRUE on success, or #VALUE! on failure.

The MAIL.NEW.MESSAGE function clears temporary attachment and message body text files and message memory. Use this function after completing a send mail operation to free temporary files and memory for the next message. Attachment files provided by the user are not deleted. Note that MAIL.NEW.MESSAGE does not alter the inbox.

## Excel Workgroup Extensions 31

To create a new, blank message, use the following function:

MAIL.NEW.MESSAGE()

Function

Description

---

SET.MAIL.MESSAGE	Sets message fields and options
MAIL.LOGON	Starts a Mail session
MAIL.FIND.NEXT	Searches for a mail message in the inbox
MAIL.DELETE.MESS AGE	Deletes the current message

---

### MAIL.RESOLVE.RECIPIENT

Macro Sheets Only

Resolves the unresolved addresses of the recipients in the recipient array and returns a recipient array of resolved recipients. For information about recipient arrays, see “Understanding Recipient and Attachment Arrays” earlier in this document.

**MAIL.RESOLVE.RECIPIENT(*recips\_array*)**

*Recips\_array* is the recipient array for the current message.

If the array contains no recipients, #N/A is returned. If the array cannot be parsed, #VALUE! is returned. If the user canceled the resolution dialog box that is displayed when resolving an ambiguous name, FALSE is returned. If an error occurred when resolving a name, the index number of the unresolvable recipient is returned.

## Microsoft Excel Function Reference

---

If a recipient cannot be resolved and you have error-trapping turned off (with `ERROR(FALSE)`), the resolution dialog is not displayed. Instead, the index number of the unresolvable recipient is returned.

You can use `GET.MAIL.MESSAGE(7)` to return the recipient array of the current message.

Resolves the names "Heidi Bardell" and "Miles Bodine" into unambiguous form suitable for use with `SET.MAIL.MESSAGE(7)`:

```
MAIL.RESOLVE.RECIPIENT({"Heikki Bardell";"Miles Bodine"}) equals
{"Heikki Bardell","MS:\SERVER1\HEIKKIBA",1;
"Miles Bodine","MS:\SERVER4\MILESB",1}
```

Note the string returned in the second column is the full unambiguous MAPI addressing syntax and varies according to your installation of Microsoft Mail.

### Function

### Description

---

`GET.MAIL.MESSAGE` Returns information about the current mail message

`SET.MAIL.RECIPIENT` Adds individual recipients to a message

`MAIL.DETAILS` Displays information about a specified recipient



---

**MAIL.SAVE.MESSAGE**

Macro Sheets Only

Saves a message in the Mail inbox.

**MAIL.SAVE.MESSAGE**(*create\_new\_logical*)

*Create\_new\_logical* is a logical value specifying whether you want to create and save a new mail message from the current message or save the current message.

- If *create\_new\_logical* is TRUE, a new Mail message is created with the contents of the current message.
- If *create\_new\_logical* is FALSE or omitted, the current Mail message is saved with changes.

The following example saves changes to the current message:

MAIL.SAVE.MESSAGE()

Function	Description
MAIL.FIND.NEXT	Finds messages in the Mail inbox
MAIL.NEW.MESSAGE	Creates a new blank message
MAIL.DELETE.MESSAGE	Deletes the current message

---

**MAIL.SEND.MESSAGE**

Macro Sheets Only

## Microsoft Excel Function Reference

---

Sends the current message. All recipients are resolved during the sending process.

**MAIL.SEND.MESSAGE()**

**MAIL.SEND.MESSAGE?()**

Returns TRUE on success, FALSE on user cancellation, and #VALUE! on failure.

Normally if either the message cannot be sent without further information or a recipient cannot be resolved and requires user input, the resolution dialog is displayed . If you have error-trapping turned off (with ERROR(FALSE) ), #VALUE! is returned if a recipient cannot be resolved or the message cannot be sent without more information.

---

If the dialog box is displayed, users can modify any item accessible from the dialog box including the subject, body text, attachments, recipient list, and sending options.

---

This function does not delete or change the current message in any way. The current message may be sent multiple times. Note that sending a message does not mark it as sent. For a message to be marked as sent, it must have been sent and received.

To send a complete message, use the following function:

MAIL.SEND.MESSAGE()

**Function**

**Description**

---

**GET.MAIL.MESSAGE** Returns information about the current message

**SET.MAIL.ATTACHMENT** Sets individual attachments to the current message

**SET.MAIL.MESSAGE** Sets contents of the current message

**SET.MAIL.RECIPIENT** Sets individual recipients to the current message

T

---

## NET.BROWSE

Macro Sheets Only

Displays dialog boxes for browsing, connecting to, and disconnecting from networks. Use the NET.BROWSE command to allow the user to easily manage network connections.

**NET.BROWSE**(*connect\_logical*, *drive\_logical*, *browse\_only\_logical*)

*Connect\_logical* is a logical value specifying whether to display the connect or disconnect dialog box provided by the network driver. Note that some network drivers, such as Microsoft Windows for Workgroups, allow both connection and disconnection from the same dialog box.

- If *connect\_logical* is TRUE or omitted, the connect dialog box is displayed.
- If *connect\_logical* is FALSE, the disconnect dialog box is displayed.

*Drive\_logical* is a logical value specifying whether to display the disk drive version or the printer version of the network connection dialog boxes.

- If *drive\_logical* is TRUE or omitted, the disk drive version of the dialog box is displayed.
- If *drive\_logical* is FALSE, the printer version of the dialog box is displayed.

*Browse\_only\_logical* is a logical value specifying whether you want to only return the name of a selected network resource or actually make the network connection.

- If *browse\_only\_logical* is TRUE and *connect\_logical* is TRUE or omitted, NET.BROWSE returns the network resource name without making the connection. This provides a useful technique for detecting what connection the user wants, and then making it with the NET.CONNECT function, since the NET.CONNECT function does not return the name of the connection just made.
- If *browse\_only\_logical* is FALSE or omitted and *connect\_logical* is TRUE or omitted, the network connection is made.

The following example allows the user to identify a desired printer and returns

Microsoft Excel Function Reference

---

the resource name as text:

NET.BROWSE(TRUE,FALSE,TRUE)

Function

Description

---

GET.NET

Returns information about the network

NET.CONNECT

Connects to a network resource

NET.DISCONNECT

Disconnects from a network resource

---

### NET.CONNECT

Macro Sheets Only

Connects to a network resource. Use NET.CONNECT to redirect a local drive letter to a network server or a port to a network printer.

**NET.CONNECT**(*net\_path\_text*, *password\_text*, *local\_name\_text*)

*Net\_path\_text* is the name of the network resource to which you want to connect, as text in the form "\\server\share".

*Password\_text* is the password for the resource, if required. Otherwise *password\_text* is omitted.

*Local\_name\_text* is the name of the local device to be connected to the network resource, of the form and range "A:" through "Z:" or "LPT1:" through "LPT9:". The local device, a disk drive letter or printer port, must be appropriate to the network resource. If *local\_name\_text* is omitted and the network resource is a file server, the next available disk drive letter is used. *Local\_name\_text* must be specified for printer connections. Attempts to connect to a local device already in use will cause an error.

The following example connects the public file server \\FINANCE\PUBLIC to the local disk drive G:

NET.CONNECT("\\FINANCE\PUBLIC", "G:")

Function

Description

GET.NET

Returns information about the network

NET.BROWSE

Displays dialog boxes for browsing, connecting to, or disconnecting from networks

NET.DISCONNECT

Disconnects from a network resource

## NET.DISCONNECT

Macro Sheets Only

Disconnects a network resource, freeing the local device for a new connection. Use NET.DISCONNECT to remove a connection from a local drive letter to a network server or a from a port to a network printer.

**NET.DISCONNECT(*device\_name*, *force\_disconnect\_logical*)**

*Device\_name* is the name of the local device you want to disconnect, as text of the form and range "A:" through "Z:" or "LPT1:" through "LPT9:".

*Force\_disconnect\_logical* is a logical value specifying whether you want to force a disconnection from a network resource that has open files or pending printer jobs, for example.

- If *force\_disconnect\_logical* is TRUE, any open resources are forced closed and the device connection is removed.
- If *force\_disconnect\_logical* is FALSE or omitted and the device has open resources, then NET.DISCONNECT returns FALSE and the device connection is retained.

The following example disconnects LPT1: from the network printer \\FINANCE\LASERJET and cancels any current print jobs:

**Microsoft Excel Function Reference**

---

NET.DISCONNECT("LPT1:;",TRUE)

**Function**

**Description**

---

GET.NET

Returns information about the network

NET.BROWSE

Displays dialog boxes for browsing, connecting to, or disconnecting from networks

NET.CONNECT

Connects to a network resource

---

**SET.MAIL.ATTACHMENT**

Macro Sheets Only

Sets individual attachments to the current message. This functions uses attachment arrays. For information on attachment arrays, see “Understanding Recipient and Attachment Arrays” earlier in this document.

**SET.MAIL.ATTACHMENT**(*index\_num*, *pathname\_text*, *label\_text*, *position\_num*)

*Index\_num* is the number of the attachment you want to set. If the attachment already exists in the attachment list, *index\_num* must be in the range 1 to GET.MAIL.MESSAGE(10). To append a new attachment to the end of the attachment list, *index\_num* must be GET.MAIL.MESSAGE(10) + 1.

*Pathname\_text* specifies the MS-DOS pathname of the attachment file, as a text string. This can be a relative pathname, although full-qualified pathnames are recommended. The file named by *pathname\_text* must exist when the attachment is specified and when the message is sent; otherwise, an error occurs.

*Label\_text* specifies the label used when the attachment icon is displayed in the message. Defaults to filename of the attachment if *label\_text* is not specified. Use a label when the attachment is stored in a temporary file that has a meaningless name.

*Position\_num* specifies the character position of the attachment within the message body. Using -1 as the position places the attachment before any body text in the message. Using -2 places the attachment after any body text. If omitted or larger than the length of the message, defaults to -2.

Returns TRUE on success and #VALUE! on failure.

To set the current attachment list using an array, use SET.MAIL.MESSAGE(7).

To append the Finance Department's January report to the list of attachments, use the following function:

```
SET.MAIL.ATTACHMENT(GET.MAIL.MESSAGE(10) + 1,"C:\FINANCE\REPORTS\
JAN.DOC","January Report",-2)
```

#### Function

#### Description

---

GET.MAIL.MESSAGE Returns information about the current message

SET.MAIL.MESSAGE Sets the contents of the current message

GET.MAIL.ATTACHM  
ENT Returns information about attachments

---

### SET.MAIL.MESSAGE

Macro Sheets Only

Sets fields of the current message. This function uses attachment and recipient arrays. For information on these, see "Understanding Recipient and Attachment Arrays" earlier in this document.

**SET.MAIL.MESSAGE(*type\_num*, *value*)**

*Type\_num* is the field that you want to set.

*Value* is the contents you want to set for the field *type\_num*. The contents of

*value* varies depending on *type\_num*:

<i>Type_num</i>	Returns
1	Message subject as text. If <i>value</i> is omitted, the message subject is cleared.
2	Message text as text, truncated to 255 characters. To set more than 255 characters, use <i>type_num</i> 3. If <i>value</i> is omitted, the message text is cleared. Setting the message body text causes all attachments with out-of-range body-positions to be moved to the end of the message body text.
3	The name of a text file containing the message body. Use this code to set the body of messages when the body is longer than 255 characters. You can delete the file after the function completes successfully, since the file contents are immediately copied to the message. <i>Value</i> is required for <i>type_num</i> =3. #VALUE! is returned if <i>value</i> is missing, if the named file cannot be read, or if not enough free memory exists to hold the text.
4	Message type as text, for example "IPM". The message type defaults to "IPM" if <i>value</i> is omitted. By convention, all message types should begin with "IPM." or "IPC." followed by application and vendor-specific information, such as "IPM.Microsoft Excel.Document Routing". IPM causes the mail to appear in the mail client's inbox list. IPC messages do not appear in the mail client's display. Periods and spaces are allowed in type names. Message type names are limited to a length of 235 characters. Type strings longer than 235 characters are silently truncated. Refer to the <i>Microsoft Mail Technical Reference</i> for more information on message types.
5	Message Conversation ID as text. Use the empty string to clear conversation ID. The conversation ID string format is determined by MAPI and may be ignored unless the transport supports conversation ID tracking.
6	The Return Receipt option. TRUE turns on receipt. Defaults



to FALSE if value is omitted.

7 Sets the recipients to the recipient array given by *value*. If *value* is omitted, the recipients list is cleared. .

8 Sets the attachments to the attachment array given by *value*. The temporary files generated by MAPI in the current attachment list are deleted. To clear all attachments, omit *value*. All attachment files named in the array must exist, or #VALUE! is returned.

To set the Return Receipt option on the current message, use the following function:

SET.MAIL.MESSAGE(5, TRUE)

To set the recipient list of the current message to the recipient array stored in the name BenefitsDepartmentManagers, use the following functionSET.MAIL.MESSAGE(7, BenefitsDepartmentManagers)

Function	Description
SET.MAIL.ATTACHMENT	Sets individual attachments to the current message
SET.MAIL.RECIPIENT	Sets individual recipients to the current message
GET.MAIL.MESSAGE	Returns information about the current message

---

## SET.MAIL.RECIPIENT

Macro Sheets Only

## Microsoft Excel Function Reference

---

Sets the name, address, and class of the recipient of the current message, or adds a recipient to the recipients array. This function uses recipient arrays. For more information on these, see “Understanding Recipient and Attachment Arrays” earlier in this document.

**SET.MAIL.RECIPIENT**(*index\_num*, *name\_text*, *address\_text*, *class\_num*)

*Index\_num* is the number of the recipient you want to set. To set information for an existing recipient, *index\_num* must be in the range 1 to GET.MAIL.MESSAGE(9). To append a new entry to the recipient list, *index\_num* must be GET.MAIL.MESSAGE(9) + 1.

*Name\_text* is the name of the recipient you want set, as text. If *name\_text* is omitted, the recipient must be specified by *address\_text*.

*Address\_text* is the address text, if needed, of the recipient you want set, as text. If *address\_text* is omitted, the recipient must be specified by *name\_text*.

*Class\_num* specifies the type of the recipient as a number: 1 = To, 2 = Cc (carbon copy), 3 = Bcc (blind carbon copy). Specifying an invalid class value results in a failure, and #VALUE! is returned.

Returns TRUE on success, or #VALUE! on failure.

You cannot omit both the *name\_text* and *address\_text* arguments.

Setting a recipient's information marks the recipient as unresolved. To set the current recipient list using an array, use SET.MAIL.MESSAGE(7).

To add Dan Williams to the CC line of your message, use the following function:

```
SET.MAIL.RECIPIENT(GET.MAIL.MESSAGE(9) + 1, "Dan Williams",2)
```

### Function

### Description

---

GET.MAIL.MESSAGE Returns information about the current message

SET.MAIL.MESSAGE Sets contents for the current mail message



