

MAPIReadMail Function

Description

Reads a MAPI message given its id.

Syntax

status = **MAPIReadMail**(*session*[, *parent*], *id*[, *flags*], *message*)

Parameters

The MAPIOpenMail function has the following parameters.

Parameter	Description
<i>session</i>	Contains a MAPI session handle returned by MAPILogon.
<i>Parent</i>	Contains the name of an OpenInSight parent window. If not specified any dialog box displayed will be application modal.
<i>Id</i>	Contains the id of the message which is to be read. Message ids are returned by MAPIOpenMail and MAPISaveMail.
<i>flags</i>	<p>Contains information about how the message should be read. It could be MAPI_ENVELOPE_ONLY\$, MAPI_SUPPRESS_ATTACH\$, MAPI_BODY_AS_FILE\$, MAPI_PEEK\$, or a combination of these flags.</p> <ul style="list-style-type: none">MAPI_ENVELOPE_ONLY\$ indicates that the message's text and file attachments should not be read.MAPI_SUPPRESS_ATTACH\$ indicates that the message's file attachments should not be read.MAPI_BODY_AS_FILE\$ indicates that the message's text should be returned as the first file attachment rather than included in the message structure.MAPI_PEEK\$ indicates that the message should not be marked as unread. <p>The insert record MAPI_EQUATES contains these flags. You may combine flags by adding them together with an addition sign (+).</p>
<i>message</i>	On output contains the message record which has been read. The message record structure is detailed in the following table.

After the function completes successfully the message parameter will contain the information read from the specified message including a list of paths to any temporary copies of file attachments. When the application is through processing the file attachments it is responsible for deleting the temporary copies with calls to the OSDelete function.

Use the [MAPI_EQUATES](#) insert record for access to the values in the following table.

Column position	Name	Purpose
1	<i>subject</i>	The subject of the message.
2	<i>from</i>	The sender of the message.
3	<i>to</i>	@VM-delimited list of the message's To: recipients.
4	<i>cc</i>	@VM-delimited list of the message's CC: recipients.
5	<i>bcc</i>	@VM-delimited list of the message's BCC: recipients.
6	<i>date</i>	The date the message was received.
7	<i>text</i>	The text of the message.
8	<i>type</i>	The type of the message.
9	<i>flags</i>	<p>The attributes of the message. It could be MAPI_UNREAD\$, MAPI_RECEIPT_REQUESTED\$, MAPI_SENT\$, or a combination of these flags.</p> <ul style="list-style-type: none">MAPI_UNREAD\$ indicates that the message has not been read.MAPI_RECEIPT_REQUESTED\$ indicates that the sender should be notified when the message is received.MAPI_SENT\$ indicates that the message has been sent. <p>The insert record MAPI_EQUATES contains these flags.</p>
10	<i>files</i>	@VM-delimited list of the names of files attached to the message.
11	<i>paths</i>	@VM-delimited list of the paths to temporary copies of files attached to the message.

12	<i>positions</i>	@VM-delimited list of the positions within the text of files attached to the message.
----	------------------	---

Returns

1 for successful execution or 0 for failure.

If function execution fails, you can retrieve a text error message by calling the `Get_Status` function. For example, to display an error message on the screen, include the following error handling routine in your code:

```
Get_Status(ErrMsg)
```

```
Msg(@window, ErrMsg)
```

See also

[Get_Status\(\)](#), [MAPILogon\(\)](#), [MAPIOpenMail\(\)](#)

Example

```
Subroutine DemoMAPIReadMail(var)
*      Display information about archived messages
$INSERT MAPI_EQUATES
Declare Function MAPILogon, MAPIOpenMail, MAPIReadMail, |
    MAPILogoff
Declare Subroutine Msg, Get_Status
flags = MAPI_ENVELOPE_ONLY$
lf     = char(10)
if MAPILogon(session) then
    if MAPIOpenMail(session, 0, "IPM.Archive", 0, ids) then
        for x = 1 to count(ids, @FM)+(ids <> "")
            if MAPIReadMail(session, 0, ids<x>, flags, message) |
                then
                    display = "From:      ":message<POS_FROM$> :lf
                    display := "Subject:   ":message<POS_SUBJECT$>:lf
                    display := "Received:  ":message<POS_DATE$>
                    Msg(@window, display)
                end else
                    Get_Status(display)
                    Msg(@window, display)
                end
            next x
        end else
            Get_Status(display)
            Msg(@window, display)
        end
    if MAPILogoff(session) else
        Get_Status(display)
        Msg(@window, display)
    end
end else
    Get_Status(display)
    Msg(@window, display)
end
return
```