

Visual Basic Programming

[Sending Email With MAPI Components in Visual Basic](#)

Contributed by [Jason Brimblecombe](#)

2002-02-15

[\[Send Me Similar Content When Posted \]](#)

[\[Add Developer Shed Headlines To Your Site \]](#)



[DISCUSS](#)



[NEWS](#)



[SEND](#)



[PRINT](#)



[PDF](#)

advertisement



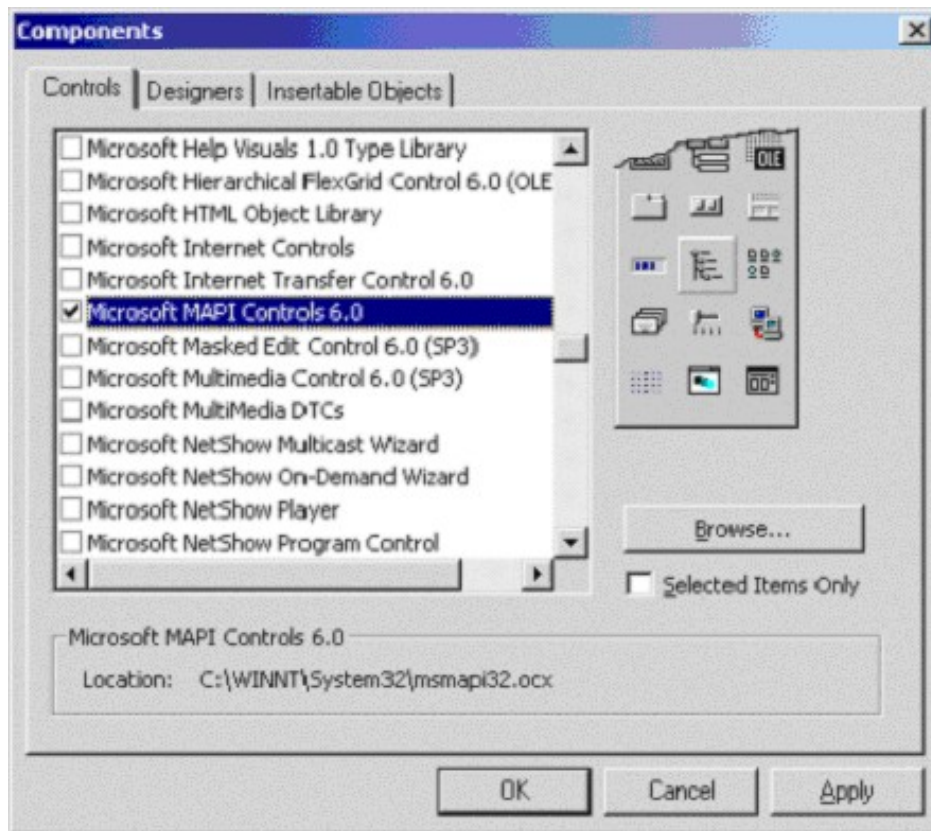
Article Index:

MAPI is an acronym for the Messaging Application Programming Interface. It's a set of core system components that can be used to automate several windows tasks. MAPI components allow developers to create mail-enabled Visual Basic applications easily. In this article Jason shows us how to use MAPI controls to read messages from our inbox, iterate through these messages, and more. MAPI is an acronym for the **M**essaging **A**pplication **P**rogramming **I**nterface. It's a set of core system components that can be used to automate several windows tasks, such as notifying a technician via email when system resources are in a critically low state, or letting a help desk manager know when all log entries have been filled, etc.

MAPI components allow developers to create mail-enabled Visual Basic (or any other language that supports COM, such as C++ or Delphi) applications amongst other things. At runtime, we can connect our application to MAPI-compliant information services through which we can both send and receive emails.

In this tutorial I will be teaching you how to send and receive e-mail through your application using the MAPI controls. You should have Visual Basic 6 installed on your machine as well as a MAPI-compliant messaging system, such as Microsoft Exchanged, Microsoft Mail or Outlook. Using the MAPI controls involves two steps: firstly establishing a MAPI session, and secondly using various MAPI properties and methods to access and manage an individual inbox; for example, to create and send a message, include a file attachment, verify the recipient's email address against the email system's address book, etc.

Let's get started on the Visual Basic project, shall we? Start by creating a new standard executable application. You will need to add the MAPI components to your project, so start by clicking project from the top menu and find the menu item components (you may also use the hotkey CTRL+T). When the components window appears, scroll down the list until you find the Microsoft MAPI controls:



Place a check in the box next to it, and then click the OK button. Two additional controls have been added to the Visual Basic toolbox; they're called MAPISession and MAPIMessages. We will be using both of these in our application.

Setting up the form

Let's setup the layout of our Visual Basic application. Our form will contain four textboxes, three labels, four command buttons, and both of the MAPI components. You should layout the controls on your form so that it looks like this (alternatively, you can see the [support material](#) for this article and download the complete project):

The image shows a screenshot of a Visual Basic form titled "Send/Receive Email Messages". The form has a blue title bar with standard Windows window controls. Below the title bar, there is a logo for "devArticles" in a white box. The form contains three text input fields labeled "From:", "To:", and "Subject:". Below these fields is a large, empty text area for the email body. At the bottom of the form, there are four buttons: a "Back" button (labeled "<< Back"), a "Forward" button (labeled "Forward >>"), a "Close Session" button, and a "Send" button. There are also two small icons on the left side of the bottom panel, one representing a printer and the other a document.

Here are the details of each component on our Visual Basic form:

Now that

Control	Name	Caption	Objective
Label	lblFrom	From:	N/A
Label	lblTo	To:	N/A
Label	lblSubject	Subject:	N/A
Textbox	txtFrom	-	Displays senders email.
Textbox	txtTo	-	Displays recipients email.
Textbox	txtSubject	-	Displays message subject.
Textbox	txtMessage	-	Displays the body of a message.
Command Button	cmdForward	Forward	Iterate to next message in inbox.
Command Button	cmdBack	Back	Iterate to previous message in inbox.
Command Button	cmdSend	Send	Sends message.
Command Button	cmdClose	Close Session	Disconnects the MAPI session.
MAPISession	MAPISession1	N/A	Signs on and establishes a MAPI session.
MAPIMessages	MAPIMessages1	N/A	Performs a variety of messaging system functions.

we've got the layout and details of the controls on our form sorted out, let's add some code to our form to interact with our MAPI compatible mail system.

Start by opening the general declarations section in the code window. Enter the following code:

Option Explicit

Dim lPosition As Long

The long integer variable that we've created in the code snippet above will be used to remember which message we are currently viewing. This variable will also be used to check whether or not we are currently viewing the last message in the inbox.

Our next step is to add some code to the Form_Load() event. Add this code to your

```
application:

Private Sub Form_Load()

'Sign on to the MAPI Session

MAPISession1.SignOn

MAPIMessages1.SessionID = MAPISession1.SessionID

MAPIMessages1.Fetch

If MAPIMessages1.MsgCount > 0 Then

'If there are messages, display the first one.

txtFrom.Text = MAPIMessages1.MsgOrigDisplayName

txtTo.Text = MAPIMessages1.RecipDisplayName

txtSubject.Text = MAPIMessages1.MsgSubject

txtMessage.Text = MAPIMessages1.MsgNoteText

'Enable the buttons to allow message navigation.

cmdBack.Enabled = True

cmdForward.Enabled = True

cmdSend.Enabled = True

Else

'No Messages? Tell the user immediately and Sign off!

MsgBox "You have no messages"

MAPISession1.SignOff

'Disable the buttons to avoid error messages

cmdBack.Enabled = False

cmdForward.Enabled = False

cmdSend.Enabled = False

End If

End Sub
```

At runtime your application will automatically sign on to a new MAPI session and check for any e-mails. If there are no e-mails, then we are informed and the session is terminated. On the other hand, if there was at least one e-mail in the Inbox, the program will show the first one in the list.

We are now going to configure the forward and backwards buttons to allow our users to navigate their way through the list of emails in their inbox. Add this code to the cmdForward_Click() event:

```
Private Sub cmdForward_Click()  
  
    'If the user has not reached the end of the list, show the next message.  
  
    If lPosition < MAPIMessages1.MsgCount Then  
  
        lPosition = lPosition + 1  
  
        'Change the Message number to the value in lPosition  
  
        MAPIMessages1.MsgIndex = lPosition  
  
        txtFrom.Text = MAPIMessages1.MsgOrigDisplayName  
  
        txtTo.Text = MAPIMessages1.RecipDisplayName  
  
        txtSubject.Text = MAPIMessages1.MsgSubject  
  
        txtMessage.Text = MAPIMessages1.MsgNoteText  
  
        'The 4 lines above will preview the current message.  
  
    End If  
  
End Sub
```

Add the following code to the cmdBack_Click() event:

```
Private Sub cmdBack_Click()  
  
    'If the user hasn't reached the beginning of the list, preview message  
  
    If lPosition > 0 Then  
  
        lPosition = lPosition - 1  
  
        'Change the Message number to the value in lPosition  
  
        MAPIMessages1.MsgIndex = lPosition  
  
        txtFrom.Text = MAPIMessages1.MsgOrigDisplayName
```

```
txtTo.Text = MAPIMessages1.RecipDisplayName
```

```
txtSubject.Text = MAPIMessages1.MsgSubject
```

```
txtMessage.Text = MAPIMessages1.MsgNoteText
```

'The 4 lines above will preview the current message.

```
End If
```

```
End Sub
```

Our program is now capable of previewing e-mail messages in the Inbox of our MAPI-compatible mailbox, such as Outlook Express.

Let's now take it to the next level by adding the ability to send e-mail. Don't let this frighten you off in any way, because it is actually just as easy to send email as it is to receive it and preview it by using the text boxes on the form. Add this code to the Click() method of the cmdSend button:

```
Private Sub cmdSend_Click()
```

'Start by telling the control that we are composing an e-mail

```
MAPIMessages1.Compose
```

'Use whatever is in the Textboxes as the information for our e-mail.

```
MAPIMessages1.RecipDisplayName = txtTo.Text
```

```
MAPIMessages1.MsgSubject = txtSubject.Text
```

```
MAPIMessages1.MsgNoteText = txtMessage.Text
```

```
MAPIMessages1.ResolveName
```

'Send the e-mail message to the Recipient

```
MAPIMessages1.Send
```

```
End Sub
```

Your application can now send emails, as well as receive them. To send an email, all we have to do is change the text in the To, Subject, and Message boxes. The email message will be sent using the email address defined in our outgoing mail application, so you don't need to change the text in the from box. There is still one more thing that needs to be added to your application. The "Close Session" button still doesn't have any code in it. Add the following code to the Click() event of the cmdClose button:

```
Private Sub cmdClose_Click()
```

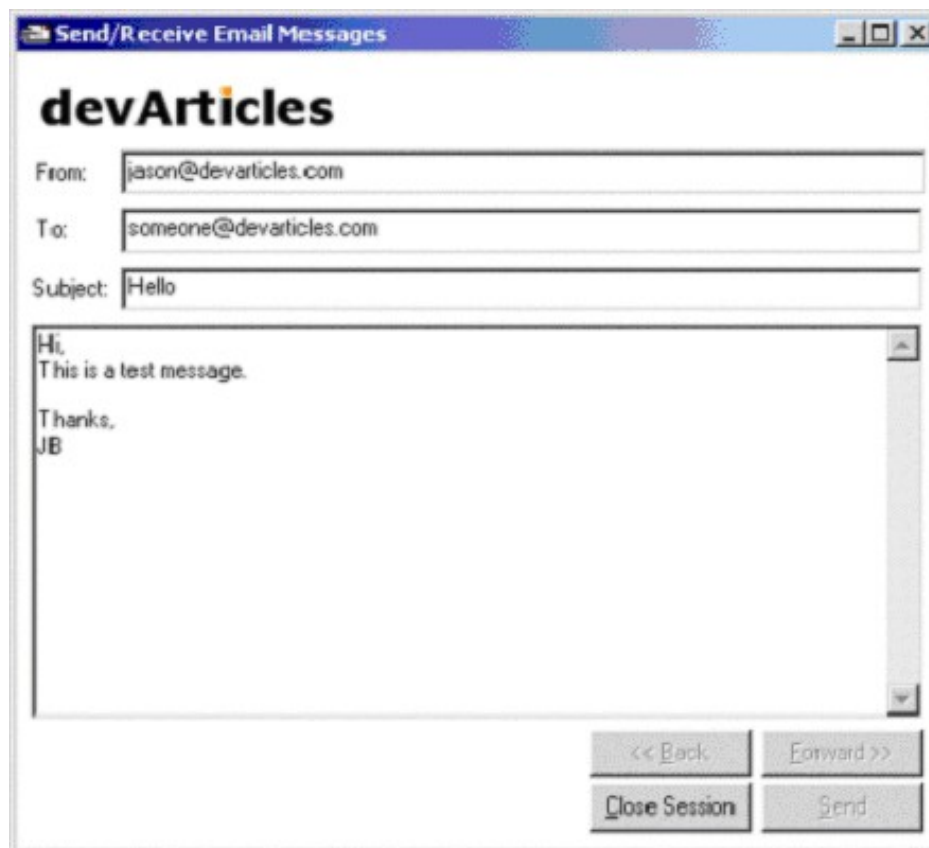
MAPISession1.SignOff

Unload Me

End Sub

When clicked, the close session button signs the user out of their e-mail client that he/she was connected to when MAPI called its SignOn method in for Form_Load() event. After we're signed out, our app will then unload itself from memory (exit the program).

Here's what our application looks like when it's running:



In this article

we've created a program that sends and receives e-mail messages using the windows built-in MAPI components. Please note that we've only looked at the very basics of using the MAPI components in this article.

There are many other features that you might want to add to this project, such as a label that displays the Message number (what message is being viewed), and even something as simple as a clear button that erases the contents of the text boxes on the form when it's clicked. You can also take a look at the links and books below if you're after more information on using the MAPI components.