Sending Email from a Visual Basic Program

Have you ever thought it would be great to be able to send email from a Visual Basic program? With Visual Basic's MAPI controls, it's a snap. These two controls let you send messages on any MAPI-compliant email system, such as Outlook and Exchange. To use the MAPI controls, you must select them in your project's Components dialog box. They are listed as "Microsoft MAPI Controls 6.0." The two controls will then appear in your toolbox; they are called MAPISession and MAPIMessages. Place one of each on a form - they are invisible at runtime.

The MAPISession control is used to establish a session, or connection, with whatever MAPI-compliant mail software is installed on the system. The control has UserName and Password properties for signing onto an email account. You can set these properties at design time or prompt the user for them in code. Then, call the control's SignOn method to establish the session. Once the session is established, the control's SessionID property returns a handle of the session.

The MAPIMessages control must be passed the handle of the MAPI session, obtained from the MAPISession control's SessionID property. Once this is done, the MAPIMessages control can be used for various tasks such as accessing messages in the InBox, saving, copying and deleting messages, and working with attachments. For this tip we are interested in creating and sending messages. This requires the following steps:

1. Call the Compose method to create a new message.
2. Put the recipient, the subject, and the body of the message in the corresponding control properties.
3. Call the Resolve method to verify the message recipient.
4. Call the Send method to send the message.

Sending a message with the MAPIMessages control does not literally send it, but puts it in the Outbox of the mail system. When the message is actually sent depends on the mail system settings. When your program is finished with mail-related activities, call the MAPISession control's SignOff method to terminate the session.

The following code demonstrates this. It assumes that the form containing the code contains TextBox controls for the various bits of information needed: user name, password, etc.

```
Private Sub SendMail_Click()

     MAPISession1.UserName = tstUserName.Text
     MAPISession1.Password = txtPassword.Text
     MAPISession1.SignOn()
     MAPIMessages1.SessionID = MAPISession1.SessionID
     MAPIMessages1.Compose()
     MAPIMessages1.RecipAddress = txtTo.Text
     MAPIMessages1.MsgSubject = txtSubject.Text
     MAPIMessages1.MsgNoteText = txtMessage.Text
     MAPIMessages1.ResolveName()
     MAPIMessages1.Send()
     MAPISession1.SignOff()

  End Sub
```

An alternate way to use the Send method is to pass an argument with the value True (the default for this argument is False, that's why it is not included in the code above).

```
MAPIMessages1.Send True
```

In this case, the Send method displays a message dialog box in which the user can enter or edit the elements of the message and then send it by clicking the Send button.

When a Visual Basic program needs to send email messages, perhaps for support or licensing issues, it's a lot nicer to integrate email support in the program rather than requiring the user to switch to their own email software.