

# Lab 9: Mail-Enabling an Application

## Lab Overview

### Objectives

After completing this lab, you will be able to:

- ♦ Start an e-mail session with the Messaging (formerly known as OLE Messaging) objects.
- ♦ Send an e-mail message with the Messaging objects.

### Scenario

In this lab, you will further extend the Main Street Market Sales Lead tool you were working with in Lab 8. You will add e-mail features so that the salespeople of Main Street Market can send e-mail to the customers identified in the Sales Lead tool directly from within the tool.

### Lab Setup

This lab builds upon the solution to Lab 8. You can either continue to work with the files you edited in Lab 8, or use the files in the \Labs\Lab09 directory.

If you need help with this lab, the solution files are in the \Labs\Lab09\Solution directory.

This demonstration shows the solution to this lab.



Estimated time to complete this lab: **60 minutes**.

## Exercises

The following exercises provide practice working with the concepts and techniques covered in this chapter.

### Exercise 1: Mail-Enabling a Visual Basic Application

In this exercise, you will add e-mail functionality to the Main Street Market Sales Lead tool.

## Exercise 1: Mail-Enabling a Visual Basic Application

In this exercise, you will add mail functionality with Messaging objects to the Main Street Market Sales Lead tool.

### ➤ Log on to the mail system and send an e-mail message

1. Open the Visual Basic project, Leads.vbp.
2. Declare an object variable for the e-mail session in the General Declarations section.
3. In the Sub procedure SendEmail, create a new e-mail session object, if it hasn't been created already.
4. If the creation of the e-mail session object is successful, log on to the e-mail system.
5. Add a new **Message** object to the e-mail session object.
6. Set the Subject text of the message.
7. Create a new **Recipient** object:
  - a. Set the **Name** property to the e-mail address of the customer selected in the lstLeads list view control. The e-mail address is SubItem 4:

```
lstLeads.SelectedItem.SubItems(4)
```
  - b. Set the **Type** property to MapiTo.

8. Send the message so that the salesperson can fill in the body before sending it.
9. Save your changes to the Leads.vbp project.

If you are unsure of the implementation of the SendMail procedure, look at this solution code:

```
Private Sub SendEmail()  
    Dim mailMsg As Object  
    Dim mailRecip As Object  
  
    'logon if not already logged on  
    If MailSession Is Nothing Then  
        Set MailSession = CreateObject("Mapi.session")  
        If Not MailSession Is Nothing Then  
            MailSession.Logon  
        End If  
    End If  
  
    Set mailMsg = MailSession.Inbox.Messages.Add  
    mailMsg.Subject = "Main St. Market Information"  
  
    Set mailRecip = mailMsg.Recipients.Add  
    mailRecip.Name = lstLeads.SelectedItem.SubItems(4)  
    mailRecip.Type = mapiTo  
  
    mailMsg.Update  
    mailMsg.Send ShowDialog:=True  
  
End Sub
```

### ➤ Log off the e-mail system when the application ends

1. Edit the Unload event of the frmLeads form.
2. If an e-mail session was started, log off.
3. Save your changes to the Leads.vbp project.

### ➤ Test the Send E-mail menu item

1. Run the Sales Lead tool.
2. Select a customer.
3. Select Send E-mail from the File menu.

### ➤ Add run-time error handling

Many errors can occur when sending an e-mail message: users can cancel the logon process, they can decide not to send the e-mail message, and so on.

- ◆ Add error handling to the SendEmail Sub procedure, and exit the Sub procedure if an error occurs.