

How to Use ASPMail

Using the component is as simple as:

1. Creating the [object](#)
2. Setting a few [properties](#)
3. Calling the [SendMail method](#)

Creating the Object

The first step to using ASPMail is to create the object. This can be done with the following code:

```
<% Set Mailer = Server.CreateObject ("SMTPsvg.Mailer") %>
```

Setting Properties

Typically you need to set five properties and call one method to define your message in ASPMail. The properties that you must set are:

FromName	The name that should be used for the Sender of the message. Example: Mailer.FromName = "Joe's Widgets Corp."
FromAddress	The email address that should be used as the sender of the message. Example: Mailer.FromAddress = "sales@joeswidgets.com"
Subject	This will be used as the Subject of the message. Example: Mailer.Subject = "Your Widget Order"
BodyText	This will be the actual message. Example: Mailer.BodyText = "Your order was processed."
RemoteHost	The SMTP server to be used for sending the message. You will want to change this to reflect the mail server that you use on your group (www13). Mailer.RemoteHost = "outbound-relay.biz.rr.com"

The method that you must call is:

AddRecipient name, email addr	This method adds a name and email address to the "To:" field of the message. Example: Mailer.AddRecipient "John Smith", "jsmith@someisp.com"
-------------------------------	--

The SendMail Method

The SendMail method takes no properties, but does return True / False to indicate Success or Failure in sending the email message.

Example 1:

```
returnCode = Mailer.SendMail
if returnCode = TRUE then
    ' Message sent sucessfully
else
    ' Message send failure
end if
```

The next example achieves the same result without using a variable for the return code.

Example 2:

```
if Mailer.SendMail then
    ' Message sent sucessfully
else
    ' Message send failure
end if
```

Error Handling

Once we've called SendMail and have checked for a send failure we may want to know what the error condition was. ASPMail provides a Property called "Response" that contains a textual description of why the message was not sent. You could use this message to inform the user or write to a log file.

Example:

```
if Mailer.SendMail then
    ' Message sent sucessfully
    response.write ("Your message was sent")
else
    ' Message send failure
    response.write ("Your message was not sent. ")
    response.write ("The error was: " & Mailer.Response)
end if
```

Completed Example

```
<%
Set Mailer = Server.CreateObject ("SMTPsvg.Mailer")
Mailer.FromName = "Joe's Widgets Corp."
Mailer.FromAddress = "sales@joeswidgets.com"
```

```
Mailer.Subject = "Your Widget Order"
Mailer.BodyText = "Your order was processed."
Mailer.RemoteHost = "outbound-relay.biz.rr.com"

Mailer.AddRecipient "John Smith", "jsmith@someisp.com"

if Mailer.SendMail then
  ' Message sent successfully
  response.write ("Your message was sent")
else
  ' Message send failure
  response.write ("Your message was not sent. ")
  response.write ("The error was: " & Mailer.Response)
end if
%>
```

Note: To CC or BCC additional recipients, add the following code (with the proper recipient names and e-mail addresses) after "Mailer.AddRecipient":

```
Mailer.AddCC "Jane Smith", janesmith@someisp.com
Mailer.AddBCC "Joe Smith", joesmith@someisp.com
```