


Learn Everything You Need to Know About HTML5

Download the free eBook: HTML5: An Introduction

[Download Now](#)

When you think ASP, think...



4 Guys From Rolla.com

SEARCH

SITE INDEX

- Recent Articles
- XML**
- All Articles
- ASP.NET Articles
- ASPFAQs.com
- Message Board
- Related Web Technologies
- User Tips!
- Coding Tips
- Search

ASP **ASP.NET** **ASPFAQs** **Message Board** **Feedback**

Looking to reduce IT costs? Learn how to cut expenses without cutting services, plus tactical approaches to controlling costs and cutting power expenses outside of the data center. Download now.

Looking to reduce IT costs? Learn how to cut expenses without cutting services, plus tactical approaches to controlling costs and cutting power expenses outside of the data center. Download now.

Published: Wednesday, July 26, 2006

Sending Email in ASP.NET 2.0

By [Scott Mitchell](#)

RESOURCES

Sections:

Book Reviews

- Sample Chapters

Commonly Asked Questions

Message Board

JavaScript Tutorials

MSDN Communities Hub

Official Docs

Security

Stump the SQL Guru!

Web Hosts

XML

Information:

Advertise

Feedback

Author an Article

Jobs

Introduction

Email serves as a ubiquitous, asynchronous notification and information distribution system. Not only in development scenarios where server-side code needs to generate an email and scuttle it off to the user, it may be destined for a user of the website, informing them of their newly created user account, reset password, or emailing them an invoice. Or it may be destined for a web developer or site administrator to handle an unhandled exception that just transpired or user feedback.

Fortunately, ASP.NET makes sending email a breeze. The .NET Framework version 1.x included a `System.Web.Mail` class that allowed programmatically sending an email with a few scant lines of code. These classes still exist in the .NET Framework version 2.0, they have been deprecated in favor of the `System.Net.Mail` namespace. (For an article on sending email in ASP.NET version 1.x, see [1.x Web Page](#) or consult www.SystemWebMail.com.)

In this article we'll look at the classes in the `System.Net.Mail` namespace and see how to send a page's code-behind class. We'll also look at specifying relay server information in `Web.config` and how it is used in some of the built-in ASP.NET server controls for sending emails (such as when a user creates a password reminder/reset). Read on to learn more!

After reading this article, be sure to check out [Sending Email in ASP.NET 2.0: HTML-Form Attachments, and Gracefully Handling SMTP Exceptions](#), where we'll look at sending HTML emails with attachments, and gracefully handling SMTP exceptions! Then mosey over to [ASP.NET 2.0: Reply-To, Priority, and Read Receipts](#) for even more great email content.

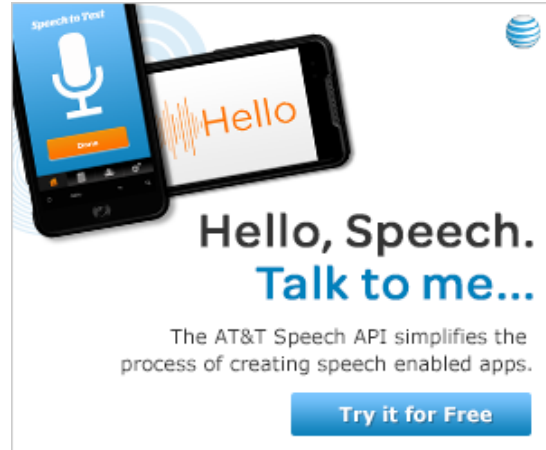
- continued -

PPC Bid
Manager

FREE TRIAL

internet.commerce

[Liability Insurance](#)
[Phone Cards](#)
[Televisions](#)
[Virtual Server](#)
[Prepaid Calling Card](#)
[Host your Site](#)
[Prepaid Phone Card](#)
[Calling Cards](#)
[Web Hosting](#)
[Business Liability](#)
[Desktop Computers](#)
[VPS Hosting](#)
[Cell Phones](#)
[IT Legal Contracts](#)



Hello, Speech.
 Talk to me...

The AT&T Speech API simplifies the process of creating speech enabled apps.

Try it for Free



Get More Out of Your Data Center

Learn how with our free eBook:

40 Ways to Make Your Data Center More Efficient

Download Now

Exploring the Classes in the `System.Net.Mail` Namespace

There are 16 different classes in the `System.Net.Mail` namespace, all related to send email to a [Protocol \(SMTP\)](#) server for delivery. The two core classes in this namespace are:

- [MailMessage](#) - represents an email message; has properties like `From`, `To`, `Subject`, `Body`
- [SmtpClient](#) - sends a specified `MailMessage` instance to a specified SMTP server.

When sending an email from an ASP.NET 2.0 page you will, typically:

1. Create a `MailMessage` object
2. Assign its properties
3. Create an instance of the `SmtpClient` class
4. Specify details about the SMTP server to use (if they're not already specified within `Web.cc`)
5. Send the `MailMessage` via the `SmtpClient` object's `Send` method

Steps 1 and 2 may be bypassed as the `SmtpClient` class's `Send` method can accept either a `Mail` representing the from, to, subject, and body contents of the email message.

The `System.Net.Mail` namespace's other classes allow for more advanced email functionality. To add attachments to an email message, to embed objects within an email, to specify SMTP server and `Exception`-derived classes for handling SMTP-specific exceptions. We'll examine using some advanced scenarios in future articles.

Providing the SMTP Server's Details

When sending an email to a friend from Outlook or GMail, the email program establishes a connection, sends the contents of the email message, along with information such as the date the email was created, the format (text or HTML, for example), the recipient(s), and so on. The relay server accepts the message, relays it to the recipient's SMTP server and sends the message. Once the message has been delivered, the recipient's mail client periodically monitors the mailbox, pulls down the message using a different protocol (such as [IMAP](#) or [POP3](#)).

Therefore, to send an email from an ASP.NET page we need to provide the `SmtplibClient` class with the hostname of the relay server, you can specify the port (typically port 25 is used when communicating your email message contents to the relay server, and authentication credentials if you have a local SMTP service installed on your web server, it may periodically monitor a particular directory for any messages that appear in that directory. You can configure whether the `SmtplibClient` class relays messages to a separate relay server or if it drops it off in a specified pickup directory through the [DeliveryMethod](#) property.

The relay server information used by the `SmtplibClient` class can be specified programmatically, through the `Web.config` file, or can be centralized in `Web.config`. To use the `Web.config` approach, add a `<system.net>` element. Then, add a `<mailSettings>` element that contains an `<smtp>` element within its `<network>` child element, like so:

```
<configuration>
  <!-- Add the email settings to the <system.net> element -->
  <system.net>
    <mailSettings>
      <smtp>
        <network>
          host="relayServerHostname"
          port="portNumber"
          userName="username"
          password="password" />
        </network>
      </smtp>
    </mailSettings>
  </system.net>

  <system.web>
    ...
  </system.web>
</configuration>
```

The `host` attribute contains the `relayServerHostname`. If you are using an external relay server, it should be something like `smtp.yourisp.com`. If the relay server's port number is something other than the default 25, specify it through the `port` attribute. Most external relay servers require authentication of some sort (in order to prevent spammers from sending their garbage through the relay). The `userName` and `password` attributes are used where username/password authentication is needed.

Only a subset of the `SmtplibClient` properties can be specified through settings in `Web.config`. To specify other properties - `EnableSsl`, `Timeout`, and so on - set them programmatically when sending the email (the steps examined earlier in this article).

Sending an Administrator Email Through a Feedback Web Page

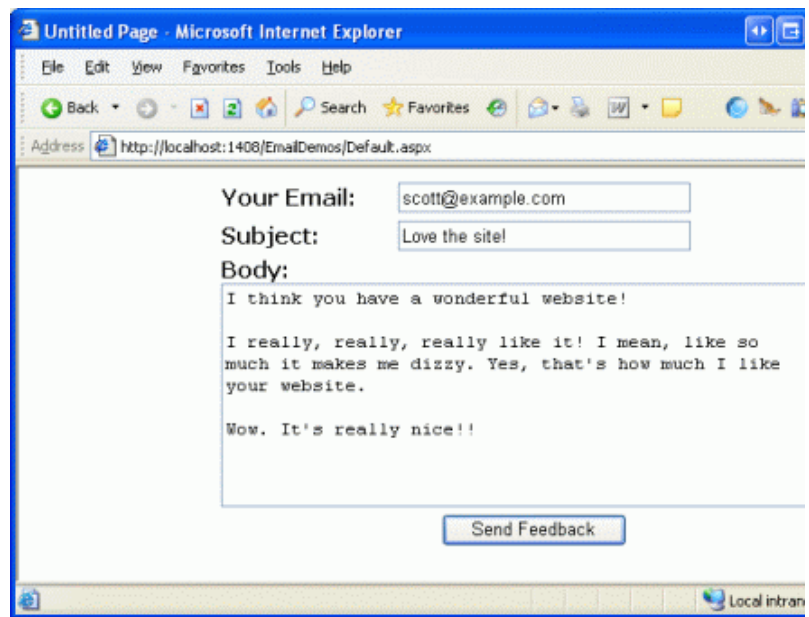
To illustrate sending an email using the `MailMessage` and `SmtplibClient` classes, I've created a simple web page. On this page the user is prompted for their email address, the subject of their feedback, and their feedback message.

```
<table border="0">
  <tr>
    <td><b>Your Email:</b></td>
    <td><asp:TextBox runat="server" ID="UsersEmail" Columns="30"></asp:TextBox>
  </tr>
  <tr>
    <td><b>Subject:</b></td>
    <td><asp:TextBox runat="server" ID="Subject" Columns="30"></asp:TextBox>
  </tr>
  <tr>
    <td colspan="2">
      <b>Body:</b><br />
      <asp:TextBox runat="server" ID="Body" TextMode="MultiLine" Columns="30">
    </td>
  </tr>
  <tr>
    <td colspan="2" align="center">
```

```

        <asp:Button runat="server" ID="SendEmail" Text="Send Feedback" />
    </td>
</tr>
</table>

```



Once the user has supplied the feedback information and clicked the "Send Feedback" button, a Click event fires. Inside the event handler, a MailMessage object is created and its To, From, Subject, and Body are set according to the information provided by the user. With the MailMessage object created and configured, the email is then sent through the SmtpClient class's Send method.

```

Protected Sub SendEmail_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    SendEmail.Click
    '!!! UPDATE THIS VALUE TO YOUR EMAIL ADDRESS
    Const ToAddress As String = "you@youreemail.com"

    '(1) Create the MailMessage instance
    Dim mm As New MailMessage(UsersEmail.Text, ToAddress)

    '(2) Assign the MailMessage's properties
    mm.Subject = Subject.Text
    mm.Body = Body.Text
    mm.IsBodyHtml = False

    '(3) Create the SmtpClient object
    Dim smtp As New SmtpClient

    '(4) Send the MailMessage (will use the Web.config settings)
    smtp.Send(mm)
End Sub

```

We didn't need to set any of the SmtpClient class's properties here in code because they have been set in the Web.config file (download the complete code at the end of this article to run this application on your computer).



Conclusion

Along with a plethora of other improvements from ASP.NET 1.x, the email sending capabilities have been moved to a new namespace, `System.Net.Mail`. In 2.0 the relay server settings can easily be moved into the `Web.config` file, as we saw in this example. Moreover, there's better security authentication. Future articles will explore more advanced email scenarios, such as: crafting HTML attachments, embedding objects within the email body, handling SMTP/relay server-related exceptions.

See [Sending Email in ASP.NET 2.0: HTML-Formatted Emails, Attachments, and Gracefully Handling Exceptions](#) for a look at sending HTML-formatted emails, emails with attachments, and handling SMTP exceptions...

Happy Programming!

- By [Scott Mitchell](#)

Attachments

- [Download the complete code samples examined in this article](#) (in ZIP format)

Suggested Readings

- [Sending Email with `System.Net.Mail`](#) (includes a C# example)
- [www.SystemNetMail.com](#) (a great set of FAQs and samples for sending email using the `System.Net.Mail` namespace)
- [Sending Email in ASP.NET 2.0: HTML-Formatted Emails, Attachments, and Gracefully Handling Exceptions](#)
- [Sending Email in ASP.NET 2.0: Reply-To, Priority, and Read Receipts](#)

[ASP.NET \[1.x\] \[2.0\]](#) | [ASPMessageboard.com](#) | [ASPFAQs.com](#) | [Advertise](#) | [Feedback](#)



Using the HTML5 History in IE 10

- Get fine-grained control over the behavior of the back button
- Improve performance without sacrificing usability



Copyright 2013 QuinStreet Inc. All Rights Reserved.

[Terms of Service](#) | [Licensing & Permissions](#) | [Privacy Policy](#)

Created with



[About the Developer.com Network](#) | [Advertise](#)