

文章编号: 2095-2163(2019)01-0252-03

中图分类号: TP311.52

文献标志码: A

# 基于.NET Framework 类库实现简单电子邮件发送系统

陈文兰

(南京市莫愁中等专业学校 电子信息系, 南京 210017)

**摘要:** 电子邮件作为 Internet 上应用最广泛的服务项目之一, 是一种便捷、安全的通信手段。 .NET Framework 类库内容丰富, 提供了多种应用程序的各类处理功能, 通过引用 .NET 类库可以方便、高效地完成程序设计工作。 本文将 .NET Framework 类库中提供的邮件发送处理功能应用于电子邮件发送系统的实现中, 大大减少了程序开发的工作量。

**关键词:** .NET Framework; 类库; 协议; 电子邮件发送系统

## Implementation of simple E-mail sending system based on .NET Framework class library

CHEN Wenlan

(Department of Electronic information, Nanjing MoChou Vocational School, Nanjing 210017, China)

**[Abstract]** As one of the most widely used services on the Internet, E-mail is a convenient and secure means of communication. The .NET Framework class library is rich in content and provides various processing functions for various applications. It is convenient and efficient to complete the programming by referring to .NET class library. In this paper, the E-mail sending and processing function provided by .NET Framework class library is applied to the implementation of E-mail sending system, which greatly reduces the workload of program development.

**[Key words]** .NET Framework; class library; protocol; e-mail sending system

### 1 .NET Framework 及类库

.NET 的核心是 .NET Framework (微软 .NET 框架体系)。 .NET Framework 有 2 个重要的组件, 一是最基本的通用语言运行时库 CLR (Common Language Runtime), 是运行时的环境, 提供程序编译、内存管理、安全性管理等功能, 是执行管理代码的代理; 二是具有多种处理功能的类库。 .NET Framework 类库是一个综合性的面向对象的 reusable 的类型集合, 包括的类型有类、接口、委托、结构和枚举等。 可以用类库来开发多种应用程序, 这些应用程序包括字符用户界面 (命令行) 或图形界面 (GUI) 应用程序, 也包括基于 ASP.NET 的应用程序。

.NET Framework 类库是以多层嵌套的“命名空间”方式来组织“类”, 或者说这些“类”分布在不同的“命名空间”中。 命名空间是用来组织和重用代码的编译单元, 也称“名称空间”或“名字空间”。 因此, 要使用命名空间所包含的类, 通常使用 using 关键字来导入该命名空间。 .NET 类库位于命名空间“System”下, System 命名空间包含了上百个类、接口、结构, 涉及了多种重要的基本处理功能。 System 命名空间中还包含许多二级命名空间, 如 System.

Net 命名空间为网络使用的多种协议提供了编程接口。 而 System.Net.Sockets 命名空间则为网络访问提供 Windows Sockets 接口的托管实现。 作为程序开发者可以定义自己的类, 从而扩展类库中的类。 当然也可以定义专用的命名空间, 这样可以避免自定义的“类”和其它“类”发生冲突。

### 2 电子邮件系统的工作过程

电子邮件系统按客户/服务器 (Client/Server, 简称 C/S) 模式工作。 发件人通过电子邮件客户端中的邮件发送程序按照 SMTP 协议先与发件人邮箱所在的邮件服务器建立连接, 再与收件人邮件服务器进行连接。 如果收件人的邮箱存在, 就将邮件传送给收件人邮件服务器, 并由收件人邮件服务器将邮件放入收件人的邮箱。 否则, 发件人信箱服务器就会退回信件, 同时通知发件人。

收件人任何时候在任何已经连接到互联网的终端 (如计算机) 上都可以检查并接收邮件。 接收邮件时, 收件人终端上运行的电子邮件客户端的邮箱访问程序会按照 POP3 协议或者 IMAP4 协议向收件人的邮件服务器提出收信请求, 只要用户输入的身份信息 (如用户名和密码) 正确, 就可以从自己的

**作者简介:** 陈文兰 (1976-), 女, 硕士, 高级讲师, 主要研究方向: C# 程序设计、ASP.NET 动态网站开发、数据库研究。

**收稿日期:** 2018-10-09

邮箱内读出邮件或下载邮件,工作过程如图 1 所示。

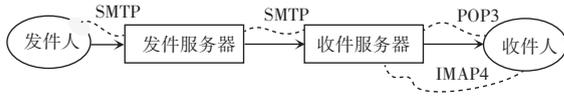


图 1 电子邮件系统的工作过程

Fig. 1 The working process of E-mail system

在电子邮件系统中,使用 SMTP 协议来发送电子邮件,并控制信件中转方式。SMTP (Simple Mail Transfer Protocol) 是一组用于由源地址到目的地址传送邮件的规则。邮件服务器执行 SMTP 协议,检查是否有邮件需要发送和接收,并负责把要发送的电子邮件发送出去,把要接收的邮件放到收件人邮箱。SMTP 协议是本文实现电子邮件发送系统的重要协议。

### 3 电子邮件发送系统实现

#### 3.1 命名空间及相关类的使用

(1) System.Net.Mail 命名空间。System.Net.Mail 命名空间提供了发送电子邮件的基本功能,此命名空间中包含用于将电子邮件发送到 SMTP 服务器所需的类或枚举,成员有 MailMessage、MailAddress、SmtpClient、Attachment、MailPriority、SmtpStatusCode 等。

(2) MailMessage 类。要成功发送一个邮件需涉及 MailMessage 类,该类用来定义邮件,包括内容、主题、收件人、发件人等信息。MailMessage 类的主要属性有 Attachment 表示电子邮件的附件; MailAddress 表示电子邮件发件人或收件人的地址; MailMessage 表示可以使用 SmtpClient 类发送的电子邮件; SmtpClient 允许应用程序使用简单邮件传输协议 SMTP 来发送电子邮件; MailPriority 指定 MailMessage 的优先级。

(3) MailAddress 类。用来定义收件人、发件人的地址。

(4) SmtpClient 类。用来发送由 MailMessage 类定义好的邮件。SmtpClient 类的主要属性有 SmtpClient, 初始化 SmtpClient 类的新实例; Credentials 验证发件人的身份; Host 获取或设置 SMTP 服务器名称或 IP 地址; Port 获取或设置 SMTP 服务器端口。Send 方法用于将电子邮件发送到 SMTP 服务器。

(5) Attachment 类。若发送邮件时,需要发送附件,就用到 Attachment 类。此类用来定义邮件的附件,要和 MailMessage 配合使用。主要属性

ContentDisposition 用来获取附件内容的 MIME 描述; ContentType 获取邮件附件内容的类型; TransferEncoding 获取或指定邮件的编码类型。

#### 3.2 发送电子邮件功能实现

以 163 的 SMTP 服务器为例,发送一个具有附件功能的电子邮件,关键代码运用如下:

```
//定义一个名为 mm 的 MailMessage 对象,对象
sclient 可以看作为一个邮件
```

```
MailMessage mm = new MailMessage();
```

```
//定义一个名为 sclient 的 SmtpClient 对象,对
象 mm 可以看作为一个邮件服务器,这里使用了
163 邮件服务器
```

```
SmtpClient sclient = new SmtpClient("smtp.163.
com");
```

```
//定义一个名为 aFile 的附件,附件所在路径为
d:\test.txt
```

```
Attachment aFile = new Attachment("d:\\test.
txt");
```

```
//定义发件人的地址
```

```
mm.From = new MailAddress(" * * * * *
* * @ 163.com");
```

```
//定义收件人的地址
```

```
mm.To.Add(new MailAddress(" * * * * @ 126.
com");
```

```
//定义抄送人的地址
```

```
mm.CC.Add(" * * * * @ qq.com");
```

```
//定义邮件的主题为“Hello”
```

```
mm.Subject = "Hello";
```

```
//定义邮件的内容关于中秋问候
```

```
mm.Body = "中秋节快乐!";
```

```
//添加一个附件
```

```
mm.Attachments.Add(aFile);
```

```
//提供一种验证方式
```

```
sclient.Credentials = CredentialCache.DefaultNetwork
Credentials;
```

```
//通过一个名字为 sclient 的邮件服务器将邮
件 mm 发出
```

```
sclient.Send(mm);
```

### 4 结束语

本文主要研究基于 .NET Framework 类库的简单电子邮件发送功能实现。在 Visual Studio .NET 2010 环境下,以面向对象且类型安全的程序设计语

(下转第 256 页)