VSTAR0.1自动发送邮件功能实现帮助文档

概述

本文档，不讨论Excel、OutLook中的自动发送邮件方法，也不讨论批处理脚本中的技巧；而仅仅从编程的角度来讨论这些问题。

如下：既有C语言实现的，也有Java、C#、Android等实现的。

# 原理

请直接参考如下网上文章：

## 1、编程实现 SMTP发邮件

<http://hi.baidu.com/cpkefzckscbfhte/item/afab3b0c06872518ebfe385a>

编程实现 SMTP发邮件

一、  
假设由发件人jmst66()发到邮箱，以下主机返回的命令都表示命令成功时返回的命令。发送过程如下：

1.连接主机smtp.163.com  
主机返回：220 163.com Anti-spam GT for Coremail System (163com[20050206])

2.第一次会话  
发送命令：EHLO (说明：发送的命令后都有换行标识"\r\n")  
返回数据：  
250-mail  
250-PIPELINING  
250-AUTH LOGIN PLAIN  
250-AUTH=LOGIN PLAIN  
250 8BITMIME

3.用户登录  
发送命令：AUTH LOGIN\r\n  
返回数据：334 dXNlcm5hbWU6 (说明：这是username的base64编码)  
发送帐号：am1zdDY2(这是jmst66的base64编码)

返回数据：334 UGFzc3dvcmQ6 (说明：这是password的base64编码)  
发送密码：xxxxxxx(这里是jmst66这个帐号的密码的base64编码)

返回数据：235 Authentication successful （登录成功）

4：发送邮箱说明  
发送命令：MAIL FROM:<  
返回数据：250 Mail OK

5.目的邮箱说明  
发送命令：RCPT TO: <  
返回数据：250 Mail OK

6.请求发送邮件  
发送命令：DATA\r\n  
返回命令：354 End data with <CR><LF>.<CR><LF>

7.发送邮件头和邮件体  
发送数据：直接是完整的数据以“<CR><LF>.<CR><LF>”结束   
返回数据：250 Mail OK queued as smtp1,wKjRC7D7f6MFoOdGa\_i7MA==.52140S2 1189584903

8.结束发送数据  
发送命令：QUIT\r\n  
接收数据：221 Bye

9.断开连接

二、  
具体程序实现简要说明如下：  
1.连接主机smtp.163.com

sockaddr\_in serverAddr;  
SOCKET sock;  
hostent \* host;  
host = gethostbyname(serverHostName.c\_str());  
if(host==NULL){  
return -1;  
}

sock = socket(AF\_INET, SOCK\_STREAM,0);  
serverAddr.sin\_addr = \*(struct in\_addr\*)host->h\_addr;  
serverAddr.sin\_family = AF\_INET;  
serverAddr.sin\_port = htons(serverPort);

if(connect(sock,(const sockaddr\*)&serverAddr,sizeof(sockaddr))==SOCKET\_ERROR)  
{  
closesocket(sock);  
return -1;  
}

//加一个判断主机返回的数据是否表示成功  
DEAL\_RETURN\_CODE(GetResponseCode(sock,220,str));

2.第一次会话  
char buffer[BUFFER\_SIZE];  
sprintf(buffer,"EHLO %s\r\n",from);//from为char数据。存储发送地址  
send(sock,buffer,strlen(buffer),0);

3.用户登录  
sprintf(buffer,"AUTH LOGIN\r\n");  
send(sock,buffer,strlen(buffer),0);  
  
//USER NAME  
//User name is coded by base64.  
EncodingBase64(username.c\_str(),buffer);//先将用户帐号经过base64编码  
strcat(buffer,"\r\n");  
send(sock,buffer,strlen(buffer),0);

//password coded by base64  
EncodingBase64(pass.c\_str(),buffer);//先将密码经过base64编码  
strcat(buffer,"\r\n");  
send(sock,buffer,strlen(buffer),0);

4：发送邮箱说明  
sprintf(buffer,"MAIL FROM:<%s>\r\n",from.c\_str());  
send(sock,buffer,strlen(buffer),0);

5.目的邮箱说明  
sprintf(buffer,"RCPT TO: <%s>\r\n",to.c\_str());  
send(sock,buffer,strlen(buffer),0);  
DEAL\_RETURN\_CODE(GetResponseCode(sock,250,str));

6.请求发送邮件  
sprintf(buffer,"DATA\r\n");  
send(sock,buffer,strlen(buffer),0);  
DEAL\_RETURN\_CODE(GetResponseCode(sock,354,str));

7.发送邮件头和邮件体，发送邮件的主题部分应该先将其格式化处理之后再发送，这里就不说明格式话的程序了。  
//DATA head  
send(sock,mailHead,strlen(mailHead),0);  
  
//DATA body  
send(sock,mailBody,strlen(mailBody),0);

8.结束发送数据  
strcpy(buffer,"QUIT\r\n");  
send(sock,buffer,strlen(buffer),0);  
DEAL\_RETURN\_CODE(GetResponseCode(sock,221,str));

9.断开连接  
closesocket(sock);

三、  
最后说明一下邮件的格式：  
邮件由两部分组成，邮件头和主体部分，  
1.邮件头,由  
Date:  
From:  
To:  
Subject:  
Message-ID:  
X-mailer:  
Mime-Version:  
等字段组成。

2.邮件体，与邮件头空一行，最后由\r\n.\r\n结尾

要详细了解邮件的结构，建议看《Visual C++网络通信协议分析与应用实现》或相关的网络编程的书籍。

如下是一封邮件经过编码后的数据：

Date:Wed, 12 Sep 2007 16:13:56 +0800  
From: <>  
To: <>  
Subject: =?gb2312?B?Sk1TVCC52NPat6Kx7c7E1cK78b2xtfey6Q==?=  
Message-ID: <>  
X-mailer: =?GB2312?B?19S8urXETWFpbGVy?=  
Mime-Version: 1.0  
Content-Type: multipart/alternative;  
         boundary="====MajorSplitTag===="  
This is a multi-part message in MIME format.

--====MajorSplitTag====  
Content-Type: text/plain;  
         charset="gb2312"  
Content-Transfer-Encoding: base64

--====MajorSplitTag====  
Content-Type: text/html;  
         charset="gb2312"  
Content-Transfer-Encoding: base64

PGh0bWw+DQo8aGVhZD4NCjxtZXRhIGh0dHAtZXF1aXY9IkNvbnRlbnQtVHlwZSIgY29udGVudD0idGV4  
dC9odG1sOyBjaGFyc2V0PWdiMjMxMiIgLz4NCjx0aXRsZT7O3rHqzOLOxLW1PC90aXRsZT4NCjwvaGVh  
ZD4NCg0KPGJvZHk+DQo8Zm9udCBzaXplPSIrMSI+DQo8cCBhbGlnbj0ibGVmdCI+1/C+tLXE1/fV36Os  
xPq6w6O6IDwvcD4NCjxwIGFsaWduPSJsZWZ0Ij4mbmJzcDsmbmJzcDsmbmJzcDsmbmJzcDvS8s7Sv6/S  
qsrVvK/E+tTaobZKTVNUobe3orHtwtvOxLXEz+C52NDFz6KjrMbk1tCw/MCou/G9scfpv/ajrM7E1cKx  
u9L908O1xLTOyv26zbG70v3Tw7XEv6/O78P7s8ajrLvy1d+78deowPvJ6sfrtcTWpMrpoaPH68T6vavL  
+beise3OxNXCtcTM4sS/o6zIq7K/1/fV36Ost6Kx7bXEICDE6qOsvu2jqMbao6mjrNKzILyw0+u45bz+  
z+C52LXEu/G9sdakyum78tXf16jA+9akyumjrMmow+i687eiy83WwTx1PkpNU1Q2NkAxNjMuQ29tPC91  
Prvy1d+9q8bkuLTTobz+tKvV5tbBIKO6IDAyNKOtODM5NzkyNDihoyA8L3A+DQo8cCBhbGlnbj0ibGVm  
dCI+1/fV37XEyNnT/tKyysfO0sPHxtq/r7XEyNnT/qOst8ezo7jQ0LvE+rbUztLDx7mk1/e1xMXkus+j  
rNKyz6PN+8T6us3E+rXEzazKwsPHxNzSu8jnvMjN+bXY1qez1s7Sw8ejoSA8L3A+DQo8cCBhbGlnbj0i  
bGVmdCI+16PE+qO6ILmk1/fLs8D7o6zN8srCyOfS4qOhIDxiciAvPg0KICC0y9bCIDwvcD4NCjxwIGFs  
aWduPSJsZWZ0Ij6+tMDxIDwvcD4NCjxwIGFsaWduPSJsZWZ0Ij5KTVNUseC8rbK/IDwvcD4NCjwvZm9u  
dD4NCjxwIGFsaWduPSJsZWZ0Ij4tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0t  
LS0tLS0tLS0gPGJyIC8+DQogIEpvdXJuYWwgb2YgTWF0ZXJpYWxzICBTY2llbmNlICZhbXA7IFRlY2hu  
b2xvZ3kgPC9wPg0KPHAgYWxpZ249ImxlZnQiPjcyIFdlbmh1YSBSb2FkLCBTaGVueWFuZyAxMTAwMTYs  
IENoaW5hIDwvcD4NCjxwIGFsaWduPSJsZWZ0Ij5UZWw6IDAyNC04Mzk3ODIwOCwgIEZheDogMDI0LTgz  
OTc5MjQ4IDwvcD4NCjxwIGFsaWduPSJsZWZ0Ij5FLW1haWw6IDxhIGhyZWY9Im1haWx0bzpKTVNUQGlt  
ci5hYy5jbiI+Sk1TVEBpbXIuYWMuY248L2E+LCA8YSBocmVmPSJtYWlsdG86Sk1TVDY2QDE2My5jb20i  
PkpNU1Q2NkAxNjMuY29tPC9hPiA8L3A+DQo8cCBhbGlnbj0ibGVmdCI+V2ViOiA8YSBocmVmPSJodHRw  
Oi8vd3d3LkpNU1QuT1JHIj5XV1cuSk1TVC5PUkc8L2E+IDxiciAvPg0KICAtLS0tLS0tLS0tLS0tLS0t  
LS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0gPC9wPg0KPC9ib2R5Pg0KPC9odG1sPg0K  
--====MajorSplitTag====

.

原文为：

尊敬的作者，您好：

     因我刊要收集您在《JMST》发表论文的相关信息，其中包括获奖情况，文章被引用的次数和被引用的刊物名称，或者获专利申请的证书。请您将所发表文章的题目，全部作者，发表的 年，卷（期），页 及与稿件相关的获奖证书或者专利证书，扫描后发送至JMST66@163.Com或者将其复印件传真至 ： 024－83979248。

作者的荣誉也是我们期刊的荣誉，非常感谢您对我们工作的配合，也希望您和您的同事们能一如既往地支持我们！

祝您： 工作顺利，万事如意！   
此致

敬礼

JMST编辑部

------------------------------------------------   
Journal of Materials Science & Technology

72 Wenhua Road, Shenyang 110016, China

Tel: 024-83978208, Fax: 024-83979248

E-mail: ,

Web: [WWW.JMST.ORG](http://www.jmst.org/)

## 2、[学习socket编程之一：用smtp协议发送邮件](http://www.cnblogs.com/jimmyhsu/archive/2005/06/16/175391.html)

这篇博客网文，原理上讲得差不多，但编程工具似乎是msdn的C#。详见：

<http://www.cnblogs.com/jimmyhsu/archive/2005/06/16/175391.html>

## 3、[[经验]](http://www.5dmail.net/bbs/forum.php?mod=forumdisplay&fid=38&filter=typeid&typeid=95) [基于SMTP协议的发送邮件WINSOCK编程](http://www.5dmail.net/bbs/thread-175925-1-1.html)

该论坛上贴出的文字，对服务器回应码，讲得较为详细。

<http://www.5dmail.net/bbs/thread-175925-1-1.html>

Winsock编程发送邮件的代码，大部分都是匿名发送的那种。很容易因为邮件服务器策略导致发送失败。现在的Smtp都需要身份验证(ESmtp),下面的代码将实现带用户名验证的winsock发送程序。  
  
具体smtp协议的一些命令参考：<http://bbs.5dmail.net/thread-175855-1-1.html>  
  
1 设置Winsock控件的通信协议（TCP)，远程主机ip,远程连接端口（通常为25）。然后连接远程主机。   
例如： Wskemail.Protocol = sckTCPProtocol  
     Wskemail.RemoteHost = "smtp.163.com"  
     Wskemail.RemotePort = 25  
     Wskemail.Connect  
  
2 如果winsock接受到的服务器的回应码为220，表明服务器端接受了客护端的连接请求。这样可以进行下一步的通信：客户端发送EHLO消息，该消息表示客户希望基于ESMTP协议的电子邮件传递。  
  
例如：  Wskemail.senddata("EHLO test"+VBcrlf)  
  
3 如果winsock接受到服务器的回应码为250，表明服务器接受了客户端的通过ESMTP发送邮件的请求。  
  这样可以进行下一步的通信：客户端发送AUTH LOGIN消息。  
  例如：Wskemail.senddata("AUTH LOGIN"+VBcrlf)  
  
4 如果winsock接收到服务器的回应码为334，表明服务器启动了一个认证过程。下一步期待用户输入用户名 。  
  例如：Wskemail.senddata(Encode64("username")+VBcrlf)  
  
5 如果winsock接收到服务器的回应码为334，表明服务器已经接受了用户名，下一步期待用户输入密码。  
例如：Wskemail.senddata(Encode64("password")+VBcrlf)  
  
6 如果winsock接收到服务器的回应码为235，表明认证成功完成。接着客户就可以发邮件了。  
  发邮件具体过程如下：  
  
  （1）客户端发送"mail from:"消息，该消息包含了发送人的email  
        例如：Wskemail.sendata("mail from:"+chr(32)+"[test@163.com"+VBcrlf](mailto:test@163.com))  
  
  （2）当客户端接收到250，则继续发送"rcpt to:"消息，它包含了收件人的邮件地址。  
        例如：Wskemail.senddata("rcpt to:"+chr(32)+"[test@163.com"+VBcrlf](mailto:test@163.com))  
  
  （3）当客户端接收到250，则继续发送“data”消息，它通知邮件服务器随后客户断将发送电子邮件数据。  
        例如：Wskemail.senddata("data"+VBcrlf)  
  
  （4）客户端如果接收到354的消息，客户端将发送电子邮件数据。电子邮件的数据包括以下字段：  
        Date,From,To,X-Mailer以及邮件正文。电子邮件以"."结束。  
  
        eg,Wskemail.senddata("Date:"+chr(32)+format(Date,"Ddd")&","&format(Date,"dd Mmm YYYY")&""&format(Time,"hh:mm:ss")&""&"-0600"+VBcrlf)  
           Wskemail.senddata("From:"+chr(32)+"test"+VBcrlf)  
           Wskemail.senddata("X-Mailer: vbemailsender"+VBcrlf)  
           Wskemail.senddata("To:"+chr(32)+"test1"+VBcrlf)  
           Wskemail.senddata("Subject:"+chr(32)+"how are you"+VBcrlf)  
           Wskemail.sendtada("How are you, this is a test email"+VBcrlf)  
           Wskemail.senddata("."+VBcrlf)  
  
   （5）如果客户端接收到250的消息，表明电子邮件数据被成功发送。此时客户端如果希望与服务器断开连接，发送"quit"消息即可。如果客户端接收到221的消息，表明服务器已经终止了基于ESMTP协议的连接。  
         例如： Wskemail.senddata("quit"+VBcrlf)

## 4、[[转帖]](http://bbs.5dmail.net/forum.php?mod=forumdisplay&fid=38&filter=typeid&typeid=94) [SMTP协议的命令和应答](http://bbs.5dmail.net/thread-175855-1-1.html)

<http://bbs.5dmail.net/thread-175855-1-1.html>

该部分对应答码，解释得较为详细。

 SMTP协议在发送SMTP和接收SMTP之间的会话是靠发送SMTP的SMTP命令和接收SMTP反馈的应答来完成的。在通讯链路建立后，发送SMTP发送MAIL命令指令邮件发送者，若接收SMTP此时可以接收邮件则作出OK的应答，然后发送SMTP继续发出RCPT命令以确认邮件是否收到，如果接收到就作出OK的应答，否则就发出拒绝接收应答，但这并不会对整个邮件操作造成影响。双方如此反复多次，直至邮件处理完毕。SMTP协议共包含10个SMTP命令，列表如下：  
　　SMTP命令命令说明  
　　HELLO＜domain＞＜CRLF＞识别发送方到接收SMTP的一个HELLO命令  
　　MAILFROM:＜reverse-path＞＜CRLF＞＜reverse-path＞为发送者地址。此命令告诉接收方一个新邮件发送的开始，并对所有的状态和缓冲区进行初始化。此命令开始一个邮件传输处理，最终完成将邮件数据传送到一个或多个邮箱中。  
　　RCPTTO:＜forward-path＞＜CRLF＞＜forward-path＞标识各个邮件接收者的地址  
　　DATA＜CRLF＞  
　　接收SMTP将把其后的行为看作邮件数据去处理，以＜CRLF＞.＜CRLF＞标识数据的结尾。  
　　REST＜CRLF＞退出/复位当前的邮件传输  
　　NOOP＜CRLF＞要求接收SMTP仅做OK应答。（用于测试）  
　　QUIT＜CRLF＞要求接收SMTP返回一个OK应答并关闭传输。  
　　VRFY＜string＞＜CRLF＞验证指定的邮箱是否存在，由于安全因素，服务器多禁止此命令。  
　　EXPN＜string＞＜CRLF＞验证给定的邮箱列表是否存在，扩充邮箱列表，也常禁止使用。  
　　HELP＜CRLF＞查询服务器支持什么命令  
　　注：＜CRLF＞为回车、换行，ASCII码分别为13、10（十进制）。  
　　SMTP协议的每一个命令都会返回一个应答码，应答码的每一个数字都是有特定含义的，如第一位数字为2时表示命令成功；为5表失败；3表没有完成。一些较复杂的邮件程序利用该特点，首先检查应答码的首数字，并根据其值来决定下一步的动作。下面将SMTP的应答码列表如下：  
　　应答码说明  
　　501参数格式错误  
　　502命令不可实现  
　　503错误的命令序列  
　　504命令参数不可实现  
　　211系统状态或系统帮助响应  
　　214帮助信息  
　　220＜domain＞服务就绪  
　　221＜domain＞服务关闭  
　　421＜domain＞服务未就绪，关闭传输信道  
　　250要求的邮件操作完成  
　　251用户非本地，将转发向＜forward-path＞  
　　450要求的邮件操作未完成，邮箱不可用  
　　550要求的邮件操作未完成，邮箱不可用  
　　451放弃要求的操作；处理过程中出错  
　　551用户非本地，请尝试＜forward-path＞  
　　452系统存储不足，要求的操作未执行  
　　552过量的存储分配，要求的操作未执行  
　　553邮箱名不可用，要求的操作未执行  
　　354开始邮件输入，以"."结束  
　　554操作失败

## 5、[原创]图解使用Telnet程序手工发送邮件

<http://haolloyin.blog.51cto.com/1177454/351256>

本文用java实现，但运行时通过telnet工具，非常清晰地显示smtp服务器和本地客户端的交互过程，有图有真相。

* **图解使用Telnet程序手工发送邮件**

我们都习惯了用邮件客户端软件或者登录到电子邮件站点进行收发邮件，现在尝试一下使用 Windows 自带的 Telnet 程序手工地发送一封简单的邮件，以此来稍微明白关于邮件发送的一些知识。

现在 E-mail 的传输一般都使用了 ESMTP （扩展的简单邮件传输协议），SMTP 中定义了 18 条命令，用于发送一封邮件。但是作为一个测试实验的话，我们只需要其中最基本的6条命令。

如下（其中的<Space>代表一个空格，<CRLF>代表回车加换行，即写代码时的 \r\n ）：

**ehlo<Space>主机名<CRLF>**

SMTP 邮件发送程序与SMTP邮件接收程序建立连接的第一条必须的命令，主机名是当前邮件发送者的主机名；

**auth<Space>login<CRLF>**

现在一般的邮箱都需要进行用户名、密码验证，而且SMTP 邮件服务器一般都使用 Login 验证方式，因此就不用考虑其他两种验证方式了；

**mail<Space>from:发送人邮箱地址<CRLF>**

从命令中很明显地看出来 mail 命令用于指定邮件发送者的邮箱地址；

**rcpt<Space>to:收件人邮箱地址<CRLF>**

同理，rcpt 命令用于指定邮件接收人的邮箱地址；

**data<CRLF>**

data 命令用于表示该命令之后的所有内容、数据都将被作为邮件的相关内容，直到遇到<CRLF>.<CRLF>标识符，即一个句号“.”结束邮件的发送内容；

**quit<CRLF>**

该命令结束邮件发送过程，关闭本机与SMTP 邮件服务器的网络连接。

在该测试中，我在mail.163.com 和 mail.sina.com 分别注册了 [test\_hao@163.com](mailto:test_hao@163.com) 和 [test\_hao@sina.cn](mailto:test_hao@sina.cn) 两个邮箱，密码都是 123456 。由于这两个邮件服务器都需要进行用户名、密码进行身份验证，而且需要将我们的用户名、密码进行 BASE64 编码之后在进行发送验证，故我们使用 Java 中的 sun.misc.BASE64Encoder 类写一个 BASE64 转码程序，源代码如下：

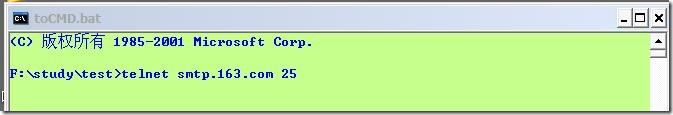
1. **import** java.io.BufferedReader;
2. **import** java.io.IOException;
3. **import** java.io.InputStreamReader;
4. //import sun.misc.BASE61Encoder;
6. **public** **class** BASE64Util
7. {
8. **public** **static** **void** main(String [] args) **throws** IOException
9. {
10. BufferedReader in = **new** BufferedReader(**new** InputStreamReader(System.in));
11. System.out.println("转码前：");
12. System.out.print("用户名 = ");
13. String userName = in.readLine();
15. System.out.print("密码 = ");
16. String pwd = in.readLine();
18. sun.misc.BASE64Encoder encoder = **new** sun.misc.BASE64Encoder();
20. System.out.println("\n转码后：");
21. System.out.println("用户名 : " + encoder.encode(userName.getBytes()));
22. System.out.println("密码  : " + encoder.encode(pwd.getBytes()));
23. }
24. }

注：由于 BASE61Encoder 类是 JDK 自带的，编译时会出现警告，我们忽略则可。现在将我们的邮箱用户名、密码进行 BASE64 编码，如下：

[](http://img1.51cto.com/attachment/201007/17/1177454_1279334542kD3V.jpg)

准备好这一些之后，可以开始发送邮件了，假设我们将邮件从 163 发送到 sina ，步骤如下（最好将以下一行一行的命令预先写在文本中，否则一旦输入错误就要重新来过了）：

1、使用 Telnet 程序连接到 163 的邮件服务器的 25 端口（该端口是默认的），如下图：

[](http://img1.51cto.com/attachment/201007/17/1177454_1279334543dTTh.jpg)

2、上面第一步完成按下 Enter 一般就会连接得上了，我们再按照上面所说的 6 条最基本的命令来发送邮件，输入ehlo hao （这个主机名自己取一个），返回一系列响应状态码，如下：

[](http://img1.51cto.com/attachment/201007/17/1177454_1279334543u85I.jpg)

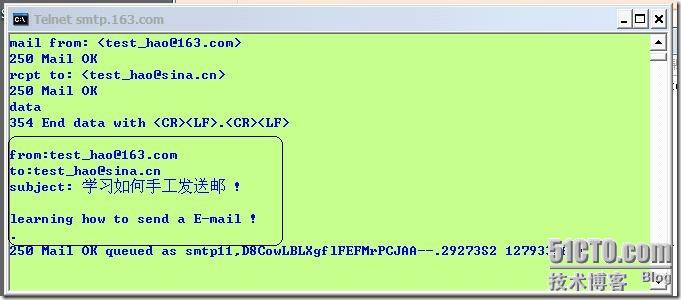
3、使用 auth 命令进行身份验证，先输入 auth login ，然后再分两步地输入转码后的用户名、密码，其中加了方框的是我们的输入，如下：

[](http://img1.51cto.com/attachment/201007/17/1177454_1279334544KPeF.jpg)

4、输入邮件发送人邮箱地址、收件人邮箱地址，如下：

[](http://img1.51cto.com/attachment/201007/17/1177454_1279334545fu5r.jpg)

6、此时可以开始输入邮件内容了，使用 data 命令，再输入内容，如下：

[](http://img1.51cto.com/attachment/201007/17/1177454_1279334546z2Uq.jpg)

7、此时，邮件发送成功了，去查看一下，如下图：

[](http://img1.51cto.com/attachment/201007/17/1177454_1279334547MGpl.jpg)

虽然发送成功了，但是上面的邮件正文不应该包含from 、to 、subject 的相关内容的，我当我从 163 发送到 qq 邮箱时，这些内容被 qq 邮件服务器解析得比较正确，格式很好，这似乎也说明不同的邮件服务器的功能服务有所差别…

使用 Telnet 手工发送邮件的过程到此结束，感受一下，觉得还真的挺麻烦的…

# 二、协议

# 编程例子

## 编程可能要参考的网站：

[**126 smtp端口**](http://www.baidu.com/link?url=pWQ4GJqjJ4zBBpC8yDF8xDhqsTSc2FUfFmM_dZ2PKduy266nTHU5hBZxL8_jozvM3ISU2knjglQv)

<http://www.126.com/help/mscan_06.htm>

<http://help.163.com/09/0219/10/52GOPOND007536NI.html>

## 一个比较复杂的工程，可以深入学习一下

Windows下纯C语言Socket、smtp发送邮

<http://download.csdn.net/download/moontalk001/4694644>

## 貌似VBA的宏：

下面的代码看明白了就可以自己写一个程序解决这个问题。

Sub AddAttachment()  
    Dim myOlApp As New Outlook.Application  
    Dim myItem As Outlook.MailItem  
    Dim myAttachments As Outlook.Attachments  
    Set myItem = myOlApp.CreateItem(olMailItem)  
    Set myAttachments = myItem.Attachments  
    myAttachments.Add "C:\Test.doc", \_  
        olByValue, 1, "Test"  
    myItem.Subject = "test"  
    myItem.Display  
End Sub

## 脚本的方法：

1. @echo off
2. :::::::::::::: 参数设置::::::::::::::
3. set from=abc@163.com
4. set user=abc
5. set pass=\*\*\*\*
6. set to=123@qq.com
7. set subj=测试
8. set mail=body.txt
9. set attach=\*.jpg
10. set server=smtp.163.com
11. set debug=-debug -log blat.log -timestamp
12. ::::::::::::::::: 运行blat :::::::::::::::::
13. blat %mail% -to %to% -base64 -charset Gb2312 -subject %subj% -attach %attach% -server %server% -f %from% -u %user% -pw %pass% %debug%

说明：很多木马会利用自身的程序截取系统敏感文件或信息发往指定的邮箱，而blat并不是木马，它小巧却有强大的发邮件功能，可不要用它做违法事，感觉和木马功能有一拼！下面先看个具体的实例（在blat同目录创建个批处理）：

是不是已经很明白了，邮件发送者地址from：abc@163.com，user指你163的用户名，pass处填你的163邮箱密码，收件人to为：123@qq.com，主题subj为：测试，邮件的正文内容mail：为body.txt里面的内容（body.txt要和blat程序同目录），邮件的附件attch为：本目录下的所有.jpg格式的图片（这点挺好，支持通配符），邮件发送[服务器](http://www.hackline.net/a/special/qyaq/server/)server为：smtp.163.com（不用设置邮件收取服务器，因为不需要本地下载邮件），考虑到传输过程[安全](http://www.hackline.net/)问题，用base64加密后传输，不易被木马截获。然后双击这个批处理，黑框框消失后你就可以进入qq邮箱验证是否发送成功了，是不是很强大！什么，还不够强大？

也许我再加点什么进去你就会承认了：第一、我把这个批处理用vbs调用，这样执行时就不会弹出窗口了，无声无息；

第二、我把这个vbs拉到启动项里面，这样每次开机都会向指定邮箱发邮件，你手机开通邮件提醒功能的话，就可以知道电脑什么时候开机了（刚刚有人动了我的电脑？）；

第三、怕本地保存的批处理暴露自己邮箱的用户名和密码？两种解决办法：  
1、申请俩小号邮箱，专门收发邮件用，呵呵；  
2、把批处理加密或者转exe文件，成为幽灵批处理；

第四、怕电脑启动时联网慢，邮件发不出去，简单！在批处理开头加上ping 127.0.0.1 -t 15，这样延迟下就可以了吧；

第五、如果加入到任务计划里面，可以实现定时发送邮件哟；第六、通过简单回复邮件，可以实现手机控制电脑操作，是不是很酷，这就需要你自己去发掘了。

# 四、问题故障及解决

## 问题1：大于65K的附件就会发送不成功？

### 描述：

C/C++ code

[?](http://bbs.csdn.net/topics/380079736#clipboardWindow)

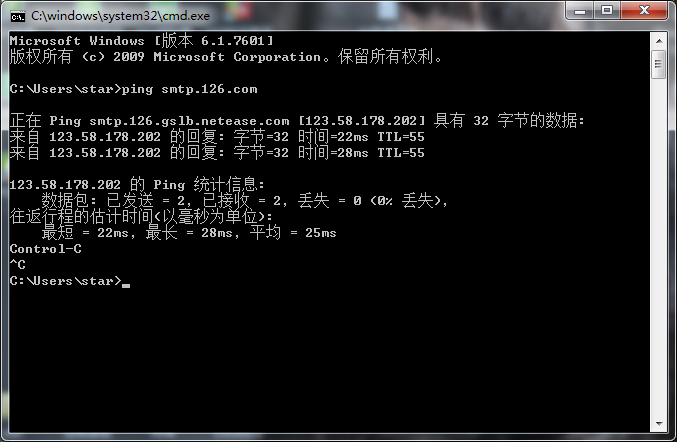
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | Char2Base64(ch,m\_pcFileBuff,40000);  rt=send(sock,ch,strlen(ch),0);  Sleep(10);  memset(ch,0,Num\*4/3+4);  Char2Base64(ch,m\_pcFileBuff,40000);  rt=send(sock,ch,strlen(ch),0);//这次发送就会不成功rt=-1； |

我写的发邮件附件的程序只能发送小于65K的附件，大于65K的附件就会发送不成功。这是什么原因啊？怎么解决？

### 解决：

问题解决了，我再每次发送的字符串末尾加上"\r\n"就能够连续的发送了。  
BASE64编码是可以使用strlen（ch）的，因为Base64编码没有'\0'字符，只要我保证ch指针末尾一定有一个'\0'就可以了

## 问题2：如何知道126邮箱的smtp服务器的ip地址？



## 问题2：mingw 找不到sys/socket.h，如何解决？

<http://blog.neohope.com/2012/01/14/mingw_socket_basic/>

需要学习参考：

MinGW中Socket基础

<http://blog.neohope.com/2012/01/14/mingw_socket_basic/>