

جامع ترین مقاله در مورد ارسال E-Mail در VB با پروتوکول SMTP

سلام به همه.
مثل اینکه مشکل فرستادن E-Mail با Visual Basic مشکل خیلی هاست.
هر چند که این مشکل توی NET برطرف شده ولی دوستانی که از VB 6.0 استفاده می کنند همچنان با این مشکل دست به گریبان اند.
گفتم از اونجایی که نون VB 6.0 رو خوردم، نمکدون نشکیم و راه استاندارد، صحیح و ۱۰۰ درصد عملی ارسال E-Mail در VB 6.0 را که تمامی برنامه های Professional از اون طریق عمل می کنند (مثل Trojan ها) برای دوستان توضیح بدم تا دوستان با مشاهده Source Code های گوناگون دچار سردرگمی نشن. که البته اکثر این کدها هم مشکل دارن و درست کار نمیدن.

همون طور که می دونید برای ارسال و دریافت داده ها در VB 6.0 از عنصر WinSock استفاده میشه. کاربران مبتدی از خود عنصر WinSock و کاربران حرفه ای همانند Trojaner ها، از توابع API WinSock که در فایل Mswsock.dll وجود دارند استفاده می کنند.
توجه: چون بیشتر کاربران از خود عنصر WinSock استفاده می کنند، بحث ما نیز در مورد شی WinSock است.

قبل از پرداختن به نحوه ارسال E-Mail برنامه نویسان عزیز باید کمی با مورد پروتوکول SMTP و نحوه ی کار آن آشنایی داشته باشند.
SMTP، مخفف عبارت Simple Mail Transfer Protocol (پروتوکول انتقال آسان Mail) می باشد که برای ارسال نامه های الکترونیکی مورد استفاده قرار می گیرد.
پروتوکول POP3 نیز که مخفف عبارت Post Office Protocol است برای دریافت E-Mail استفاده می شود.
با ترکیب و بکار گیری این دو پروتوکول با همدیگر، ما یک سیستم جامع و کامل مدیریت پیغام های الکترونیکی خواهیم داشت.
و اما...

از اونجایی که پروتوکول، زبان برقراری ارتباط محسوب میشه و هر پروتوکول پیغام های مخصوص به خودش رو داره، پروتوکول SMTP نیز از این قائده مستثنا نبوده و برای برقراری ارتباط با این پروتوکول باید از پیغام های ثابت و استاندارد استفاده کنیم.
البته جالبیه این پیغام ها در اینه که بسیار به زبان انسان شباهت دارند.
چهار پیغام اصلی که در این پروتوکول مورد استفاده قرار می گیرد به شرح زیر است:

HELO
این پیغام مشخص کننده ی نوع Mail Server ای است که آدرس E-Mail شخص گیرنده در آن قرار دارد. به عنوان مثال: MAIL.HOMAIL.COM
نکته ی بسیار بسیار مهم: هر MAIL SERVER ای قابلیت پاسخگویی به پیغام های شما را ندارد و این بسته به نوع عرضه ی سرویس POP3 بر روی آن MAIL SERVER است.
اگر MAIL SERVER، سرویس POP3 را به طور رایگان عرضه کند، به پیغام های شما پاسخ خواهد داد در غیر اینصورت برنامه شما بدون خطا کاری انجام نمی دهد.
توجه: سرویس POP3 سایت Yahoo و Hotmail پولی است ولی احتمالاً سرویس POP3 سایت MSN مجانی است.

MAIL FROM:
این پیغام آدرس E-Mail شما را مشخص می کند.
توجه: این پیغام می تواند حاوی هر آدرس E-Mail ای باشد. مهم نیست که این آدرس E-Mail وجود داشته باشد یا نه. مثلاً می توانید از طرف خود شخص Bill Gates و با آدرس وی برای یک نفر Mail بفرستید. (از این بهتر نمیشه!)

RCPT TO:
این پیغام مشخص کننده ی آدرس E-Mail فردی است که E-Mail باید برای او ارسال شود.
توجه: آدرس E-Mail شخص گیرنده حتماً باید وجود داشته باشد.

DATA
قسمت اصلی پیغام که شامل متن نامه، فایل های ضمیمه شده و تگ های HTML می باشد.

این چهار پیغام، پیغام های اصلی و مورد نیاز برای ارسال E-Mail بود.

در قبال ارسال هر پیغام به سمت Mail Server و دریافت آنها توسط آن، Mail Server نیز پیغام هایی را برای آگاهی از نتیجه ی پیغام، برای شما ارسال می کند. این پیغام ها يك عدد سه رقمی است که یکی از مقادیر زیر می باشد:

۲۲۰: پیغام ها به صورت ناقص دریافت شدند ولی به هر حال دریافت شده اند.

۲۵۰: پیغام شما با موفقیت دریافت شد. منتظر دریافت پیغام بعدی.

۳۵۴: آماده دریافت داده ها.

۲۲۱: ارتباط Mail Server با شما قطع شده است.

در صورت بروز خطا در ارسال داده ها، Mail Server با اعداد ۵۰۰، ۵۵۰ و ... پاسخ می دهد. به عبارت دیگر هر عدد دیگری بیانگر بروز خطاست.

خوب تا اینجا خسته نباشید...

این مقدمات پروتوکول SMTP بود.

کار اصلی از حالا شروع میشه که باید این داده ها را به Mail Server ارسال کنیم.

من برنامه ی خودم رو که برای ارسال E-Mail نوشته ام و در يك Trojan نیز از اون استفاده کردم در اینجا به طور جامع و کامل توضیح می دهم.

در بالای فرم اصلی ارسال ایمیل، خطوط زیر را که بیانگر انواع پیغام هایی است که ما برای برای برقراری ارتباط با Mail Server از آنها استفاده می کنیم، تعریف کنید:

[code]

```
Private Enum State
```

```
CONNECT
```

```
HELO
```

```
FROM
```

```
RCPTTO
```

```
DATA
```

```
DOT
```

```
QUIT
```

```
End Enum
```

```
Private m_State As State
```

```
Private m_strEncodedFiles As String
```

[/code]

CONNECT: به WinSock می گوید که به SMTP سرور مربوطه وصل شو.

HELO: نمایانگر Mail Server است.

FROM: آدرس ایمیل فرستنده.

RCPTTO: آدرس ایمیل گیرنده.

DATA: داده های ارسالی که می تواند شامل متن، HTML و فایل های ضمیمه باشد.

DOT: از این مشخصه برای مشخص کردن پایان E-Mail استفاده می شود.

QUIT: مشخصه ای که به سرور می گوید کاربر را DC کن.

توجه: متغیر m_strEncodeFiles برای ارسال فایل های ضمیمه مورد استفاده قرار می گیرد.

يك List Box برای نمایش اسامی فایل های ضمیمه شده، به فرم برنامه اضافه کنید.

دو تا Command Button نیز به فرم اضافه کنید. یکی برای انتخاب و اضافه کردن فایل به List Box و یکی دیگه برای حذف فایل انتخاب شده از List Box.

نام دکمه ها را به ترتیب به cmdAdd و cmdRemove تغییر بدید.

يك Common Dialog Box هم برای انتخاب فایل مورد نظر، به فرم اضافه کنید و اسمش را به C1

تغییر دهید.

حالا کد زیر را در روال Click دکمه cmdAdd بنویسید:

[code]

```
Private Sub cmdAddFile_Click()
```

```
C1.DefaultExt = "All files (*.*)|*.*"
```

```
C1.Filter = "All files (*.*)|*.*"
```

```
C1.ShowOpen
```

```
If C1.FileName <> "" Then
```

```
List1.AddItem C1.FileName
End If
End Sub
[/code]
```

کد زیر را نیز به روال Click دکمه cmdRemove اضافه کنید:

```
[code]
Private Sub cmdRemove_Click()
On Error Resume Next
List1.RemoveItem List1.ListIndex
End Sub
[/code]
```

حالا يك دکمه ديگه جهت ارسال E_Mail به فرم اضافه کنید و اسم اون رو به cmdSend تغيير بدید.
توجه: بقيه عناصر مورد نیاز در طول توضیحات برنامه معرفي مي شوند.
کد زیر را در روال Click دکمه cmdSend بنویسید:

```
[code]
Private Sub cmdSend_Click()
Dim I As Integer
Dim strServer As String, ColonPos As Integer, lngPort As Long
شمارش تعداد فایل هایی که باید برای گیرنده ارسال شود.
For I = 0 To List1.ListCount - 1
تابع UUEncode برای ایجاد Packet مربوط به فایل ضمیمه شده به کار مي رود و در آخر برنامه نوشته شده است
m_strEncodedFiles = m_strEncodedFiles & UUEncodeFile(List1.List(I)) &
vbCrLf
Next I
```

```
Text Box يك TxtServer است که آدرس SMTP سرور را شامل مي شود
strServer = Trim(txtServer)
متغير ColonPos برای فهمیدن اینکه آیا کاربر از يك پروکسي سرور برای برقراري ارتباط با
Mail Server استفاده مي کند یا خير استفاده مي شود. مثال: 127.0.0.1:4020
ColonPos = InStr(strServer, ":")
اگر ColonPos برابر با صفر بود يعني هيچ پروکسي برای اتصال وجود ندارد.
If ColonPos = 0 Then
خط زیر به عنصر WinSock مي گوید که به سرور وصل شو. پورت شماره ۲۵، پورت پيش فرض برای
SMTP سرور است.
```

```
Winsock1.Connect strServer, 25
Else
به دست آوردن عدد پروکسي که برای برقراري ارتباط با Mail Server استفاده مي شود.
LngPort = CLng(Right$(strServer, Len(strServer) - ColonPos))
strServer = Left$(strServer, ColonPos - 1)
Winsock1.Connect strServer, lngPort
End If
```

وضعیت جاری (در حالت اتصال)

```
m_State = CONNECT
End Sub
[/code]
```

خوب. تا اینجاي کار که آسون بود. اما کار اصلي در روال DataArrival عنصر WinSock انجام ميشه.
ارسال پيغام ها و همچنین دریافت پاسخ ها از Mail Server توسط اين روال انجام مي گيرد.
البته کد اين روال يك مقدار طولانيه ولي فهمش آسونه.

```
[code]
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
پاسخ ارسال شده توسط سرور در اين متغير نگهداري مي شود
Dim strServerResponse As String
کد معادل پاسخ سرور مثلاً: 220، 250
Dim strResponseCode As String
داده اي که قرار است ارسال شود
Dim strDataToSend As String
```

پیغام های واقعی که شامل ضمایم، پیغام ها و ... هستند

```
Dim strMessage As String
```

با دستور زیر به Winsock می گوییم که پاسخ های سرور را دریافت کن

```
Winsock1.GetData strServerResponse
```

استخراج تنها سه رقم سمت چپ از پاسخ دریافت شده

```
strResponseCode = Left(strServerResponse, 3)
```

اگر هر کدام از اعداد زیر در پاسخ سرور دریافت شد، نمایانگر این است که داده ها با موفقیت ارسال شده اند.

```
If strResponseCode = "250" Or strResponseCode = "220" Or  
strResponseCode = "354" Then  
Select Case m_State  
Case CONNECT
```

تغییر وضعیت جاری

```
m_State = HELO
```

Text Box يك txtSender است که آدرس E_Mail فرستنده در آن قرار می گیرد

```
strDataToSend = Trim$(txtSender)
```

استخراج قسمت مشخصه اصلی ایمیل

```
strDataToSend = Left$(strDataToSend, InStr(1, strDataToSend, "@") - 1)
```

عمل ارسال داده ها به Mail Server با ارسال پیغام HELO آغاز می شود

```
Winsock1.SendData "HELO " & strDataToSend & vbCrLf  
Case HELO
```

تغییر وضعیت جاری

```
m_State = FROM
```

ارسال پیغام MAIL FROM: به Mail Server که مشخص کننده ی آدرس فرستنده است

```
Winsock1.SendData "MAIL FROM:" & Trim$(txtSender) & vbCrLf  
Case FROM
```

تغییر وضعیت جاری

```
m_State = RCPTTO
```

Text Box يك txtRecipient است که مشخص کننده ی آدرس شخص گیرنده است

```
Winsock1.SendData "RCPT TO:" & Trim$(txtRecipient) & vbCrLf  
Case RCPTTO
```

تغییر وضعیت جاری

```
m_State = DATA
```

ارسال پیغام DATA به Mail Server برای آنکه بگوئیم ارسال داده ها آغاز شده است

```
Winsock1.SendData "DATA" & vbCrLf  
Case DATA
```

تغییر وضعیت جاری

```
m_State = MAIL_DOT
```

txtSenderName مشخص کننده ی نام فرستنده است

```
Winsock1.SendData "From:" & txtSenderName & "<" & txtSender & ">" &  
vbCrLf
```

TxtRecipientName مشخص کننده ی نام دریافت کننده است

```
Winsock1.SendData "To:" & txtRecipientName & "<" & txtRecipient & ">" &  
vbCrLf
```

از اونجایی که بنده با مد روز پیش می رم يك Text Box هم به نام txtReplyTo به برنامه اضافه کردم

```
If Len(txtReplyTo.Text) > 0 Then  
Mail ارسل داده ی مشخص کننده ی موضوع  
Winsock1.SendData "Subject:" & txtSubject & vbCrLf  
Winsock1.SendData "Reply-To:" & txtReplyToName & "<" & txtReplyTo &  
">" & vbCrLf & vbCrLf  
Else  
Winsock1.SendData "Subject:" & txtSubject & vbCrLf & vbCrLf  
End If
```

و در نهایت txtMessage که مشخص کننده ی متن E_Mail است

```
strMessage = txtMessage & vbCrLf & vbCrLf & m_strEncodedFiles
```

کار ما با متغیر m_strEncodeFiles تمام شده است و محتویات این متغیر در متغیر strMessage قرار گرفته است بنابراین آن را برابر با Null قرار می دهیم تا حافظه اصلی اشغال شده به سیستم بازگردانده شود

```
m_strEncodedFiles = ""
    ارسال محتویات اصلی Mail (متن و ضمایم)
Winsock1.SendData strMessage & vbCrLf
    بازگرداندن حافظه اصلی اشغال شده به سیستم
strMessage = ""
    توجه مهم: برای مشخص کردن پایان عملیات ارسال E_Mail، حتما باید یک علامت نقطه نیز در پایان
    به Mail Server ارسال شود
Winsock1.SendData "." & vbCrLf
Case DOT
    تغییر وضعیت جاری
m_State = QUIT
    ارسال پیغام QUIT به Mail Server برای قطع ارتباط کاربر از آن مورد نیاز است
Winsock1.SendData "QUIT" & vbCrLf
Case QUIT
    بستن WinSock
Winsock1.Close
End Select
Else
    در ارسال داده ها اشکالی به وجود آمده (خطا در ارسال)
    پس ارتباط جاری WinSock با Mail Server باید قطع شود
Winsock1.Close
    اگر سرور به پیغام QUIT، پاسخ نداد پس حتما اشکالی به وجود آمده است
If Not m_State = QUIT Then
MsgBox "SMTP Error: " & strServerResponse, vbInformation, "SMTP Error"
Else
    و اگر پاسخ داد، پس عملیات ارسال با موفقیت انجام پذیرفته است
MsgBox "Message Sent Successfully.", vbInformation
End If
End If
End Sub
[/code]
```

و اما...

اگر عنصر WinSock در حین عملیات خود به مشکلی برخورد کرد، می توان خطاهای احتمالی را با روال WinSock_Error، کنترل کرد:

```
[code]
Private Sub Winsock1_Error(ByVal Number As Integer, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile
As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
MsgBox "Winsock Error Number " & Number & vbCrLf & Description,
vbExclamation, "Winsock Error"
End Sub
[/code]
```

و در پایان، کدهای تابع UUEncodeFile:

توجه: الگوریتم زیر، یک الگوریتم استاندارد به منظور استخراج تنها اطلاعات لازم از یک فایل باینری و کم کردن حجم آن است.

```
[code]
Public Function UUEncodeFile(strFilePath As String) As String
Dim intFile As Integer      ' هندل فایل
Dim intTempFile As Integer  ' فایل موقت
Dim lFileSize As Long       ' اندازه فایل
Dim strFilename As String   ' نام فایل
Dim strFileData As String   ' محتویات فایل
Dim lEncodedLines As Long   ' تعداد خطوط کد شده در فایل
Dim strTempLine As String   ' رشته موقت
```

```

Dim i As Long          شماره حلقه
Dim j As Integer        شماره حلقه
Dim strResult As String

استخراج نام فایل
strFilename = Mid$(strFilePath, InStrRev(strFilePath, "\") + 1)
قرار دادن يك نشانه در ابتدای نام فایل
strResult = "begin 664 " + strFilename + vbCrLf
به دست آوردن اندازه فایل
lFileSize = FileLen(strFilePath)
lEncodedLines = lFileSize \ 45 + 1
ایجاد يك بافر برای قرار دادن ۴۵ قسمت از فایل در آن (بحث مربوط به ذخیره داده ها در فایل های
باینری)
strFileData = Space(45)
intFile = FreeFile
Open strFilePath For Binary As intFile
For i = 1 To lEncodedLines
خواندن ۴۵ قسمت اول فایل
If i = lEncodedLines Then
از آنجا که معمولا آخرین خط داده در فایل های باینری برابر با ۴۵ نیست، بنابراین باید اندازه بافر تخصیص
داده شده را تغییر بدهیم
strFileData = Space(lFileSize Mod 45)
End If
داده های فایل را خوانده و در بافر قرار می دهیم
Get intFile, , strFileData
اولین مشخصه را به رشته کد شده اضافه می کنیم که مشخص کننده ی تعداد مشخصه های به کار
رفته در رشته کد شده است. البته بیشتر وقت ها از کاراکتر M استفاده میشه
strTempLine = Chr(Len(strFileData) + 32)
If i = lEncodedLines And (Len(strFileData) Mod 3) Then
اگر خط آخر فایل، با موفقیت پردازش شد و طول داده پردازش شده بر عدد ۳ قابل پردازش نباشه، يك یا
دو فضای خالی به فایل اضافه می کنیم.
strFileData = strFileData + Space(3 - (Len(strFileData) Mod 3))
End If
For j = 1 To Len(strFileData) Step 3
هر ۸ بیت را به ۶ بیت تبدیل می کنیم

يك بایت
strTempLine = strTempLine + Chr(Asc(Mid(strFileData, j, 1)) \ 4 + 32)
دو بایت
strTempLine = strTempLine + Chr((Asc(Mid(strFileData, j, 1)) Mod 4) *
16 + Asc(Mid(strFileData, j + 1, 1)) \ 16 + 32)
سه بایت
strTempLine = strTempLine + Chr((Asc(Mid(strFileData, j + 1, 1)) Mod
16) * 4 + Asc(Mid(strFileData, j + 2, 1)) \ 64 + 32)
چهار بایت
strTempLine = strTempLine + Chr(Asc(Mid(strFileData, j + 2, 1)) Mod
64 + 32)
Next j
فضاهای خالی را با کاراکتر "``" جایگزین می کنیم
strTempLine = Replace(strTempLine, " ", "")
خط کد شده را به بافر پردازش شده اضافه می کنیم
strResult = strResult + strTempLine + vbCrLf
خالی کردن بافر
strTempLine = ""
Next i
Close intFile
اضافه کردن نشانه پایان
strResult = strResult & "``" & vbCrLf + "end" + vbCrLf
نتیجه نهایی
UUEncodeFile = strResult

```

```
End Function  
[/code]
```

خسته نباشید.

این روش استانداردترین روش استاندارد E-Mail هست که در اکثر برنامه ها استفاده میشه. البته هرکس نسبت به نیاز خودش می تونه امکاناتی رو به این کد اضافه یا کم کنه.

نکته بسیار بسیار مهم:

این نکته بسیار مهمه که این برنامه تنها بر روی Mail Server هایی کار می کنه که سرویس POP3 رو به صورت رایگان عرضه می کنند.

من به شخصه قبلا از Hotmail استفاده می کردم ولی سرویس Hotmail POP3 الان پولی شده. فکر می کنم که MSN رایگان باشه.

به هر حال راه بسیار مناسب دیگری هست که در موقع مقتضی به دوستان عرض می کنم.

توجه: اکثر برنامه های ارسال E-Mail که ملاحظه کردید به خاطر عدم ایجاد صحیح Packet ایمیل کار نمی کنند.

کاملترین مرجع آشنایی با استانداردهای WEB و PC، مراجع RFC هستند.

RFC مربوط به E-Mail، RFC822 است.

موفق باشید.

با تشکر فراوان از مطالعه این مقاله.

بهروز راد