

24.3.3 MimeMessage

Die Klasse `MimeMessage` (gb.mime) wird stets zusammen mit den Klassen `Mime` und `MimePart` verwendet

- um den Quelltext einer EMail aus Header und Body zu erzeugen oder
- den Quelltext einer EMail in Header und Body zu zerlegen.

Die Klasse kann erzeugt werden. Eine neue EMail vom Typ `MimeMessage` wird so erzeugt:

```
Dim hMimeMessage As MimeMessage
hMimeMessage = New MimeMessage ( [ Contents As String ] )
```

- Der Inhalt des optionalen Parameters 'Contents' bestimmt den Text in der Mime-Nachricht.
- Fehlt der Parameter, so wird eine leere Mime-Nachricht erzeugt.

24.3.3.1 Eigenschaften

Die Klasse `MimeMessage` verfügt über diese Eigenschaften, die in der folgenden Tabelle beschrieben werden.

Eigenschaft	Datentyp	Beschreibung
Sender	String	Setzt den Absender der EMail oder gibt den Absender zurück.
To	String[]	Setzt den Empfänger der EMail oder gibt den Empfänger zurück.
Cc	String[]	Setzt weitere Empfänger der EMail in Kopie oder gibt diese optionalen Empfänger zurück.
Bcc	String[]	Setzt weitere, jedoch als versteckt deklarierte Empfänger der EMail oder gibt diese optionalen Empfänger zurück.
ReplyTo	String	Setzt das ReplyTo-Header-Feld oder gibt das ReplyTo-Header-Feld zurück.
Subject	String	Setzt den Betreff der EMail oder gibt den Betreff zurück.
Id	String	Setzt die ID der EMail oder gibt die ID der EMail zurück.
Body	MimePart	Gibt den Inhalt der EMail zurück. Der Inhalt ist Text zum Beispiel in der Kodierung text/plain oder text/html, in den optional multimediale Objekte in unterschiedlichen Kodierungen eingefügt werden können. Diese Eigenschaft ignoriert jedoch Anhänge.
Headers	.MimeMessage.Headers	<code>MimeMessage.Headers[fieldname]</code> gibt ein über 'fieldname' definiertes Header-Feld einer EMail zurück. Die virtuelle Klasse 'MimeMessage.Headers' kann wie ein Schreib/Lese-Array verwendet werden.

Tabelle 24.3.3.1.1 : Eigenschaften der Klasse `MimeMessage`

Kommentar:

- Über ausgewählte Eigenschaften der Klasse `MimeMessage` haben Sie direkten Schreib-Lese-Zugriff auf spezielle Felder im Header wie To, From, Subject oder Body, die in jeder EMail enthalten sind.
- Andere Header-Felder – wie zum Beispiel das Datum in einer EMail – können Sie über die Eigenschaft `Headers["fieldname"] = Headers["Date"]` auslesen oder setzen.
- Den Body-Teil können Sie nur als Text vom Typ `MimePart` (→ Kapitel 24.3.2) auslesen oder setzen.

Beispiel 1 – Inhalt von Header-Feldern festlegen

Mit diesem Quelltext-Ausschnitt setzen Sie den Wert einer Kopfzeile (Header-Feld) mit dem Bezeichner 'FieldName':

```
Dim hMimeMessage As MimeMessage
Dim sString As String
hMimeMessage.Headers [ FieldName As String ] = sString
```

Beispiel 2 – Inhalt von Header-Feldern auslesen

So geben Sie den Wert einer Kopfzeile (Header-Feld) mit dem Bezeichner 'FieldName' zurück:

```
Dim hMimeMessage As MimeMessage
Dim sString As String
sString = hMimeMessage.Headers [ FieldName As String ]
```

24.3.3.2 Methoden

Die Klasse *MimeMessage* besitzt nur eine Methode. Die Funktion *ToString* setzt eine EMail aus ihren Teilen Header und Body zusammen:

```
Function ToString ( ) As String
```

24.3.3.3 Projekt

Mit dem folgenden Quelltext-Ausschnitt können Sie nicht nur den Inhalt ausgewählter Header-Felder sondern auch Body-Sub-Header-Felder in einem EMail-Quelltext auslesen und anzeigen. Außerdem erhalten Sie Informationen zu den EMail-Anhängen, sofern diese vorhanden sind. Den Informationen wird eine Übersicht zur Struktur der untersuchten EMail vorangestellt:

```
Public Sub btnShowMimeMessageDetails_Click()
    Dim hMimeMessage As New MimeMessage
    Dim hMimePart, hMimePart2 As MimePart
    Dim sMessage, sEncoding As String
    Dim k As Integer = 1

    hMimeMessage = New MimeMessage(txAMonitor.Text)

    $sBodyType = Scan(hMimeMessage.Part.Headers["Content-Type"], ";*")[0]

    sMessage = ("D E T A I L S M I M E - M E S S A G E")
    txAMonitor.Insert(sMessage & gb.NewLine)
    txAMonitor.Insert(String$(String.Len(sMessage), "-") & gb.NewLine)
    txAMonitor.Insert(gb.NewLine)

    ' Parse mimemessage ...
    txAMonitor.Insert("+ " & $sBodyType & gb.NewLine)
    ParsePart(hMimeMessage.Body, True)
    ParsePart(hMimeMessage.Part, False)

    ' Show details ...
    txAMonitor.Insert(String$(80, "-") & gb.NewLine)
    txAMonitor.Insert(gb.NewLine)

    txAMonitor.Insert("CONTENT-TYPE EMAIL = " & Scan(hMimeMessage.Headers["Content-Type"], ";*")[0] & gb.-
    NewLine)
    txAMonitor.Insert("DATE = " & hMimeMessage.Headers["Date"] & gb.NewLine)
    txAMonitor.Insert("FROM = " & hMimeMessage.Sender & gb.NewLine)
    txAMonitor.Insert("RETURN-PATH = " & hMimeMessage.Headers["Return-Path"] & gb.NewLine)
    txAMonitor.Insert("SUBJECT = " & hMimeMessage.Subject & gb.NewLine)
    txAMonitor.Insert("TO = " & hMimeMessage.To & gb.NewLine)
    If hMimeMessage.Cc Then
        txAMonitor.Insert("CC = " & hMimeMessage.Cc & gb.NewLine)
    Endif
    ' txAMonitor.Insert("BCC = "; hMimeMessage.Bcc ' Blind means hidden ...
    txAMonitor.Insert("MESSAGE-ID = " & hMimeMessage.Id & gb.NewLine)
    txAMonitor.Insert("MIME-VERSION = " & hMimeMessage.Headers["MIME-Version"] & gb.NewLine)

    hMimePart = New MimePart
    hMimePart = hMimeMessage.Part
    hMimePart2 = New MimePart
    hMimePart2 = hMimeMessage.Part

    If Scan(hMimeMessage.Headers["Content-Type"], ";*")[0] Like "*/mixed" Then ' : Attachment number >= 1
        txAMonitor.Insert(gb.NewLine)
        sMessage = ("D E T A I L S F O R A T T A C H M E N T")
        txAMonitor.Insert(sMessage & gb.NewLine)
        txAMonitor.Insert(String$(String.Len(sMessage), "-") & gb.NewLine)
        txAMonitor.Insert(gb.NewLine)
    Endif

    For Each hMimePart2 In hMimePart
        If hMimePart2.Disposition = "attachment" Then
            txAMonitor.Insert(("Attachment ") & Str(k) & ":" & gb.NewLine)
            Inc k
        Endif
    Next
End Sub
```

```

txaMonitor.Insert("Disposition = " & hMimePart2.Disposition & gb.NewLine)
Select Case hMimePart2.ContentEncoding
  Case 0
    sEncoding = "Default"
  Case 1
    sEncoding = "7Bit"
  Case 2
    sEncoding = "8Bit"
  Case 3
    sEncoding = "Binary"
  Case 4
    sEncoding = "Base64"
  Case 5
    sEncoding = "QuotedPrintable"
  Case 6
    sEncoding = "UUEncode"
End Select

txaMonitor.Insert("ContentEncoding = " & sEncoding & gb.NewLine)
txaMonitor.Insert("ContentType = " & hMimePart2.ContentType & gb.NewLine)
txaMonitor.Insert("FileName = " & hMimePart2.FileName & gb.NewLine)
txaMonitor.Insert(gb.NewLine)
Endif
Next

btnShowMimeMessageDetails.Enabled = False

End ' btnShowMimeMessageDetails_Click()

```

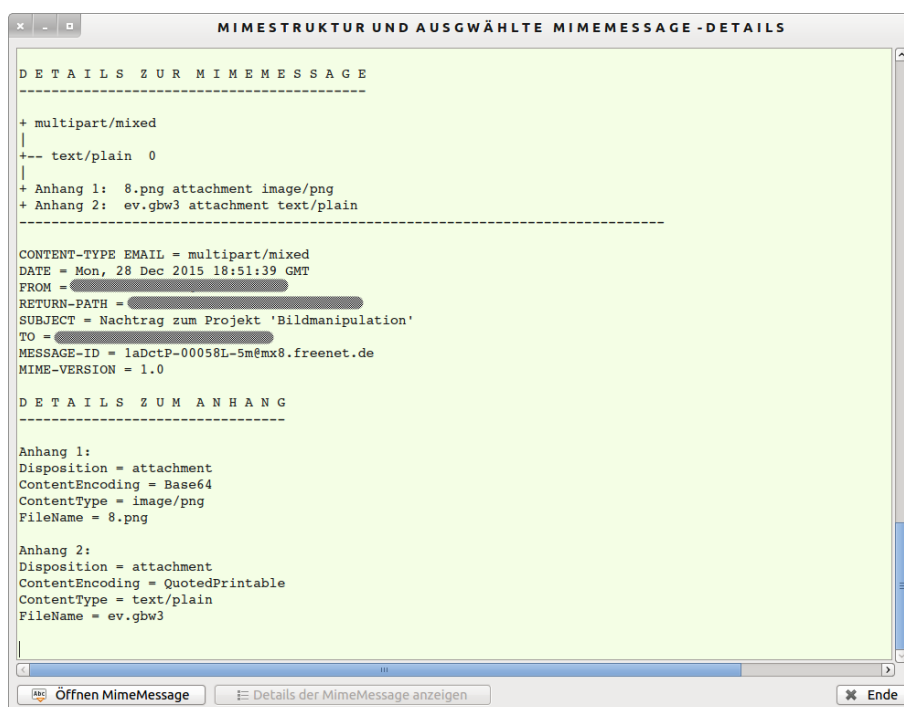


Abbildung 24.3.3.3.1: Anzeige ausgewählter Details in einer EMail als MimeMessage