

24.4.2 SMTP Projekt 2

In diesem Kapitel wird Ihnen ein SMTP-Client vorgestellt, dessen Besonderheit darin besteht, dass Sie eine EMail nicht direkt, sondern mit dem Standard-EMail-Programm Ihres Systems verschicken. Ihr Vorteil: Die *Verbindungssicherheit* und die erforderliche *Authentifikationsmethode* stellen Sie für das genutzte EMail-Konto im Standard-EMail-Programm des Systems ein.

Das Projekt nutzt die Methode `SendMail(...)` der Klasse `Desktop` der gleichnamigen Komponente, die bereits im → Kapitel 15.1.1 beschrieben wurde.

24.4.2.1 Projekt – SMTP-Client

Mit Hilfe der Methode `Desktop.SendMail(..)` können Sie eine EMail aufbereiten und an den Standard-EMail-Client des Systems übergeben:

```
SendMail(TO As String[], CC As String[], BCC As String[], Subject As String, Body As String, Attachment As String)
```

- TO ist eine Liste der Empfänger (String-Array)
- CC ist eine weitere Liste (String-Array) von Empfängern mit CC für *carbon-copy*
- BCC ist eine weitere Liste (String-Array) von Empfängern mit BCC für *blind-carbon-copy*
- Subject ist der Betreff der EMail
- Body ist der Text in der EMail
- Attachment ist der Pfad zur Anhang-Datei, wobei nur eine Datei akzeptiert wird. Die Ursache ist darin zu sehen, dass der Parameter 'Attachment' in der Methode `SendMail(...)` vom Typ `String` ist! Diese Beschränkung können Sie umgehen, in dem Sie zum Beispiel alle Anhänge in ein Archiv verpacken.

Die Angabe einer (aktiven) EMail-Adresse (TO) ist zwingend notwendig. Alle weiteren Argumente sind optional. Es reicht völlig aus, wenn Sie in Ihrem Gambas-Programm die notwendigen und optionalen Parameter programm-intern bereitstellen oder die Daten mit einem Formular erfassen:

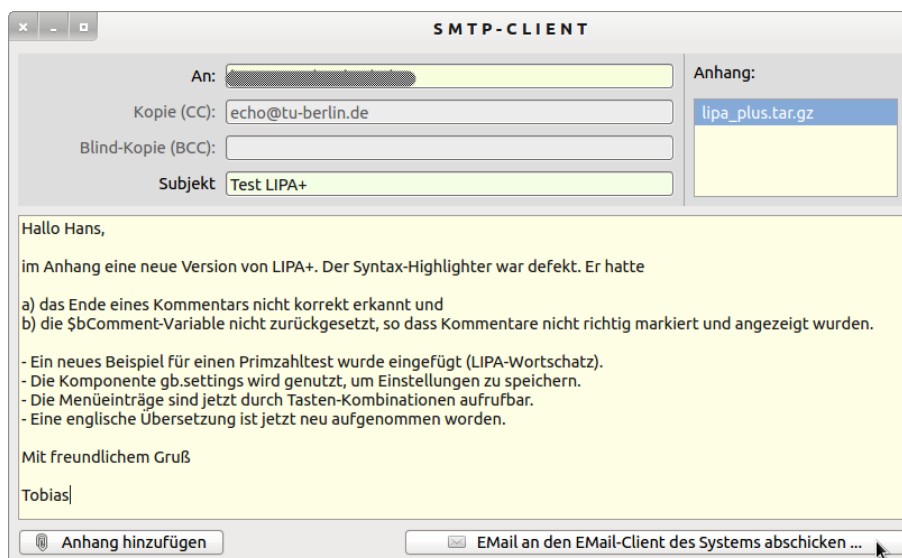


Abbildung 24.4.2.1.1: GUI: SMTP-Client

Nach dem Abschicken der EMail werden die Daten an den Standard-EMail-Client des Systems übergeben. Die folgenden Hinweise beziehen sich auf den EMail-Client *Thunderbird*:

- Der Anhang wird übertragen und korrekt angezeigt. Achtung: Nicht alle EMail-Clients übernehmen den Anhang korrekt.
- Sie können im Programm nur eine Datei als Anhang auswählen.
- Die Empfängerlisten werden nicht als komma-getrennte Liste in den EMail-Client übernommen, sondern in separaten, einzelnen Zeilen!
- Es hat sich gezeigt, dass jeweils nur ein Empfänger aus den Listen CC und BCC korrekt eingetragen wird. Alle weiteren erscheinen als normaler Empfänger.

Das liegt m.E. daran, dass intern das Konsolen-Programm *xdg-email* benutzt wird, um eine EMail an den bevorzugten EMail-Client zu schicken. Im Programm *xdg-email* wird ein feiner Unterschied zwischen mehreren Adressen bei TO und jeweils einer Adresse bei CC und BCC gemacht.

```
xdg-email [--utf8] [--cc address] [--bcc address] [--subject text] [--body text ] [--attach file]
[ mailto-uri | address(es) ]
```

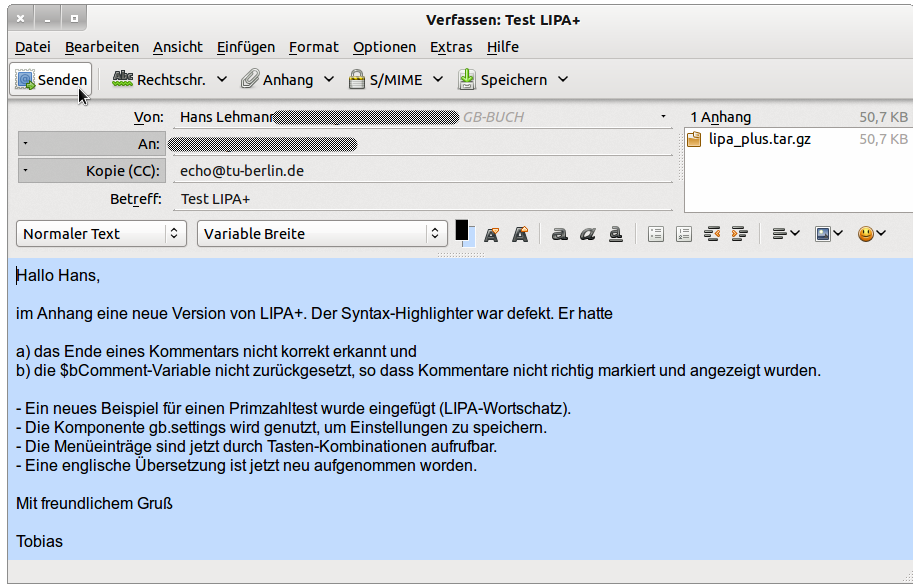


Abbildung 24.4.2.1.2: Standard-EMail-Programm im System mit den übergebenen Daten

24.4.2.2 Projekt-Quelltext

Der Quelltext scheint sehr umfangreich zu sein – aber das täuscht, weil umfangreiche Prüfungen der erfassten Argumente für die Methode `Desktop.SendMail(...)` sicherstellen sollen, dass nur valide Daten übergeben werden:

```
' Gambas class file

Private sEMailAnhangPfad As String

Public Sub Form_Open()
    SetFormPropertyts
End ' Form_Open()

Public Sub btnAddAttachment_Click()

' Es wird nur ein Anhang akzeptiert, denn der Parameter Attachment in SendMail(...) ist vom Typ String!
If lboxAttachments.Count = 1 Then
    lboxAttachments.Remove(0)
EndIf
Dialog.Title = "EMail-Anhang-Auswahl-Dialog"
Dialog.Path = User.Home
If Dialog.OpenFile() Then Return
HSplit1.Layout = [75, 25]
sEMailAnhangPfad = Dialog.Path
lboxAttachments.Add(" " & File.Name(sEMailAnhangPfad))
lboxAttachments.Index = lboxAttachments.Count - 1

End ' btnAddAttachment_Click()

Public Sub btnEMailToDesktopClient_Click()
    Dim sToList, sCCList, sBCCList, sSubject, sEMailBody As String
    Dim sAttachment, sMessage, sEMPattern, sElement As String
    Dim aSendToI, aSendCC1, aSendBCC1 As New String[]
    Dim aSendTo, aSendCC, aSendBCC As New String[]

    sEMPattern = "^[\w-]+(?:\.[\w-]+)*@(?:[\w-]+\.)+[a-zA-Z]{2,7}$" ' Muster zur Syntax-Prüfung
    sEMPattern = "^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,7}$"
    -----
' EMail-Adressen der Empfänger festlegen (mindestens eine EMail-Adresse )
```

```

aSendTo1 = New String[]
If txtTo.Text Then
    sToList = Replace(txtTo.Text, Chr(32), "")
    If InStr(sToList, ",") Then
        aSendTo1 = Split(sToList, ",")
    Else
        aSendTo1.Add(sToList)
    Endif ' InStr(sToList, ",") ?
Else
    Message.Warning("Die EMail hat keinen Empfänger!")
    txtTo.SetFocus
    Return
Endif ' txtTo.Text ?

aSendTo = New String[]
For Each sElement In aSendTo1
    If sElement Match sEMPattern Then
        aSendTo.Add(sElement)
    Else
        sMessage = Message.Warning("Mindestens eine Empfänger-EMail-Adresse ist nicht korrekt!")
        txtTo.SetFocus
        Return
    Endif
Next

'-----
Option: Zusätzliche EMail-Adressen der Empfänger festlegen - CC (CarbonCopy)
aSendCC1 = New String[]
If txtCC.Text Then
    sCCList = Replace(txtCC.Text, Chr(32), "")
    If InStr(sCCList, ",") Then
        aSendCC1.Add(Split(sCCList, ",")[0]) ' Nur die erste Adresse wird übernommen
    Else
        aSendCC1.Add(sCCList)
    Endif ' InStr(sCCList, ",") ?

    aSendCC = New String[]
    For Each sElement In aSendCC1
        If sElement Match sEMPattern Then
            aSendCC.Add(sElement)
        Else
            sMessage = "Mindestens eine CC-Empfänger-EMail-Adresse<br>(CarbonCopy) ist nicht korrekt!"
            Message.Warning(sMessage)
            txtCC.SetFocus
            Return
        Endif
    Next
Endif ' txtCC.Text ?

'-----
Option: Zusätzliche, versteckte EMail-Adressen der Empfänger festlegen - BCC (BlindCarbonCopy)
aSendBCC1 = New String[]
If txtBCC.Text Then
    sBCCList = Replace(txtBCC.Text, Chr(32), "")
    If InStr(sBCCList, ",") Then
        aSendBCC1.Add(Split(sBCCList, ",")[0]) ' Nur die erste Adresse wird übernommen
    Else
        aSendBCC1.Add(sBCCList)
    Endif ' InStr(sBCCList, ",")

    For Each sElement In aSendBCC1
        If sElement Match sEMPattern Then
            aSendBCC.Add(sElement)
        Else
            sMessage = "Mindestens eine BCC-Empfänger-EMail-Adresse<br>(BlindCarbonCopy) ist nicht korrekt!"
            Message.Warning(sMessage)
            txtBCC.SetFocus
            Return
        Endif
    Next
Endif ' txtBCC.Text ?

'-----
If txtSubject.Text Then
    sSubject = txtSubject.Text
Else
    Message.Warning("Die EMail hat keinen Betreff!")
    txtSubject.SetFocus
    Return
Endif ' txtSubject.Text ?

'-----
If txtMailBody.Text Then
    sEMailBody = txtMailBody.Text
Else
    Message.Warning("Die EMail enthält keine Nachricht!")
    txtMailBody.SetFocus
    Return

```

```

Endif ' txaMailBody ?

' -----
sAttachment = sEmailAnhangPfad
' -----

Try Desktop.SendMail(aSendTo, aSendCC, aSendBCC, sSubject, sEmailBody, sAttachment)
If Error Then
    sMessage = "Fehler bei der Übergabe der EMail"
    sMessage &= gb.NewLine & "an das Desktop-Standard-EMail-Programm!"
    sMessage &= gb.NewLine & "Fehler-Text: " & Error.Text
    Message.Error(sMessage)
Endif

End ' btnEmailToClient_Click()

Private Sub SetFormPropertyts()
    FMain.Center
    FMain.Arrangement = Arrange.Vertical
    FMain.Expand = False
    FMain.Margin = True
    FMain.Spacing = True
    FMain.Utility = True

' Container HSplit
HSplit1.Layout = [100, 0] ' Man sieht nach dem Programmstart die Box für die Anhänge zunächst nicht

' Container VBox links im Container HSplit
vboxLeft.Margin = True ' Fester Abstand zwischen VBox-Rand und den 4 inneren HBoxen
vboxLeft.Spacing = True ' Fester Abstand zwischen den 4 inneren HBoxen

hboxTo.Spacing = True ' Fester Abstand zwischen TextLabel und TextBox in der HBox (Empfänger)
txlTo.W = 168 ' Feste Breite des TextLabels → Gilt für alle 4 TextLabel
txlTo.Alignment = Align.Right ' Text-Ausrichtung auf dem TextLabel ist "rechts"
txbTo.Expand = True ' Die Weite der Textbox passt sich dynamisch an die Container-Weite an

hboxCC.Spacing = True
txlCC.W = 168
txlCC.Alignment = Align.Right
txbCC.Expand = True

hboxBCC.Spacing = True
txlBCC.W = 168
txlBCC.Alignment = Align.Right
txbBCC.Expand = True

hboxSubject.Spacing = True
lblSubject.W = 168
lblSubject.Alignment = Align.Right
txbSubject.Expand = True

' Container rechts im Container HSplit
' panAttachment ist ein Panel mit spezifischen Eigenschaftswerten (ohne Rand, Anordnung vertikal)
panAttachment.Border = Border.None ' Panel ohne Rand
panAttachment.Arrangement = Arrange.Vertical ' Vertikale Anordnung der 2 inneren HBoxen
panAttachment.Margin = True ' Fester Abstand zwischen Panel und den zwei inneren HBoxen
panAttachment.Spacing = True ' Fester (vertikaler) Abstand zwischen den zwei inneren HBoxen

lblAttachment.Alignment = Align.TopLeft ' Textanzeige in der linken, oberen Ecke
hboxListBox.Expand = True ' Maximale Ausdehnung
lboxAttachments.Expand = True ' Maximale Ausdehnung
lboxAttachments.ScrollBar = Scroll.Vertical
lboxAttachments.Mode = Select.Single

' Steuerelement TextArea
MailBody.Expand = True ' Füllt automatisch die Fläche, die HSplit und HBox (unten) frei lassen

' Container HBox unten
hboxBottom.Height = 24
hboxBottom.Margin = False
hboxBottom.Spacing = True
panSpace.Expand = True ' Schiebt die 2 Button an den linken bzw. an den rechten Rand des Containers

btnEmailToDesktopClient.Text = " EMail an das Desktop-Standard-EMail-Programm schicken"

txbTo.Clear
txbCC.Clear
txbBCC.Clear
txbSubject.Clear
txaMailBody.Clear
txaMailBody.Wrap = True
End ' SetFormPropertyts

```