

24.2.3.1 Projekt 1 – FTP-Client

Die Klasse FTPClient (gb.net.curl) stellt einen FTP-Client zur Verfügung, der das Herunterladen von Dateien von einem FTP-Server, das Hochladen von Dateien auf einen FTP-Server und das Senden von FTP-Befehlen zu einem FTP-Server ermöglicht. Diese Funktionalität sollte im Projekt demonstriert werden. Die Programmoberfläche stellt einerseits die Sicht auf ein lokales Verzeichnis und die darin enthaltenen Dateien dar und andererseits u.a. auf ein Remote-Verzeichnis und dessen Dateien. Als Steuerelemente werden für die lokale Sicht eine DirView und eine FileView verwendet. Die Remote-Verzeichnisse auf dem FTP-Server werden auf eine TreeView abgebildet. Für die Anzeige der Datei-Namen (mit passendem Symbol) und ausgewählten Datei-Eigenschaften in einem ausgewählten Remote-Verzeichnis wird eine GridView eingesetzt:

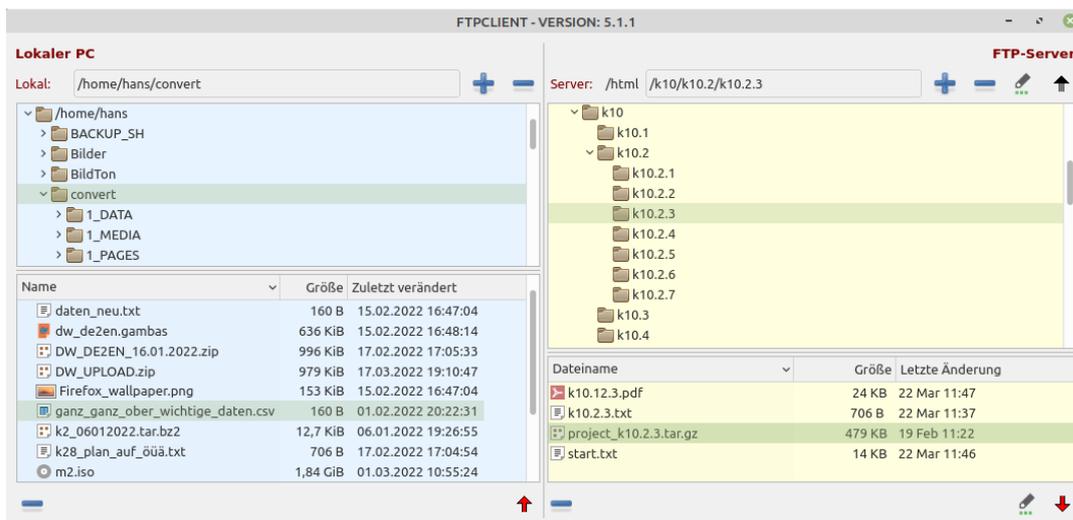


Abbildung 24.2.3.1.1: GUI – Programm FTP-Client1

Die Funktionalität des FTP-Clients lässt sich vereinfacht so beschreiben:

PC: Sie können

- ein Verzeichnis erzeugen (+), ein Verzeichnis löschen (-),
- eine Datei löschen (-) und
- eine oder mehrere markierte Dateien in das ausgewählte Verzeichnis auf dem FTP-Server hochladen (↑).
- Sie können alternativ eine oder mehrere Dateien mit Drag&Drop in das ausgewählte Verzeichnis auf dem FTP-Server hochladen!

FTP-Server: Sie können

- ein Verzeichnis erzeugen (+), ein Verzeichnis löschen (-),
- ein Verzeichnis umbenennen,
- schrittweise in das angegebene (Basis-)Verzeichnis wechseln (im o.a. Fall ist das /html),
- eine Datei löschen (-),
- eine Datei umbenennen und
- eine oder mehrere markierte Dateien in das ausgewählte lokale Verzeichnis herunterladen (↓).
- Sie können alternativ eine oder mehrere Dateien mit Drag&Drop in das ausgewählte lokale Verzeichnis herunterladen!

Für den Test des Programms wurde der FTP-Server *vsFTPD* lokal installiert und passend konfiguriert. Unter <https://gambas-buch.de/doku.php?id=k24:k24.15:start> finden Sie im Kapitel 24.15 in einem Exkurs eine ausführliche Beschreibung der Installation und Konfiguration des sicheren FTP-Servers.

Die Idee zu diesem Gambas-Projekt stammt von Jorge Carrión (2013). Er hat jedoch sein Programm auf der Basis der Klasse FTPClient mit einem anderen Ansatz (GUI) und neuen Akzenten weiterentwickelt. Sie finden das finale Projekt von Jorge jetzt unter <https://gitlab.com/shordi/gbftp>.

24.2.3.1.1 Hinweis zu FTP und FTPS

Die Klasse FTPClient (gb.net.curl) bietet gegenwärtig (23.03.2022) nur **unverschlüsseltes FTP** und ist daher unsicher! Bitte wechseln Sie zu FTP über TLS (FTPS).

24.2.3.1.2 Verbindungsdaten

Der Dialog zur Festlegung der FTP-Verbindungsdaten erfasst die URL des FTP-Servers, den FTP-Benutzernamen, dessen FTP-Passwort und den Basis-Pfad auf dem FTP-Server:



Abbildung 24.2.3.1.2: Dialog 1 – FTP-Verbindungsdaten

24.2.3.1.3 Quelltext Dialog – FTP-Verbindungsdaten

Der Dialog zum Einlesen der FTP-Verbindungsdaten wird im Hauptprogramm aufgerufen und ist erfreulich kurz:

```
' Gambas class file
' ● Unverschlüsseltes FTP ist unsicher. Bitte wechseln Sie zu FTP über TLS.
' ● Unencrypted FTP is insecure. Please switch to FTP over TLS.

Private cFTPAccountData As Collection

Public Sub _call(sTitel As String) As Collection

    Me.Text = sTitel

'-- Returns if one of the two buttons was clicked!
'-- The return value is set in the Me.Close() call and indicates whether the dialog was aborted or not.
    If Me.ShowModal() = 0 Then
        Return Null
    Else
        If txbFTPServerURL.Text And If txbFTPUserName.Text And If txbFTPUserPassword.Text And \
        If txbFTPServerInitialPath.Text Then
            cFTPAccountData = New Collection
            cFTPAccountData["FTPServerURL"] = txbFTPServerURL.Text
            cFTPAccountData["FTPUsername"] = txbFTPUserName.Text
            cFTPAccountData["FTPPassword"] = txbFTPUserPassword.Text
            If txbFTPServerInitialPath.Text Not Ends "/" Then txbFTPServerInitialPath.Text &= "/"
            cFTPAccountData["FTPServerInitialPath"] = txbFTPServerInitialPath.Text
            Return cFTPAccountData
        Else
            Message.Error(("The FTP account data are not complete!"))
            Return Null
        Endif
    Endif
End

Public Sub Form_Open()
    txbFTPServerURL.Text = MSettings.AppSettings["FTP/ServerURL"]
    txbFTPUserName.Text = MSettings.AppSettings["FTP/Username"]
    txbFTPUserPassword.Text = MSettings.AppSettings["FTP/Password"]
    txbFTPServerInitialPath.Text = MSettings.AppSettings["FTP/InitialPath"]
End

Public Sub bConnect_Click()
    Me.Close(1)
End

Public Sub bCancel_Click()
    Me.Close(0)
End

Public Sub Form_Close()
    If rbtnRemember.Value = True Then
```

```

MSettings.AppSettings["FTP/ServerURL"] = txtFTPServerURL.Text
MSettings.AppSettings["FTP/Username"] = txtFTPUserName.Text
MSettings.AppSettings["FTP/Password"] = txtFTPUserPassword.Text
If txtFTPServerInitialPath.Text Not Ends "/" Then txtFTPServerInitialPath.Text &= "/"
MSettings.AppSettings["FTP/InitialPath"] = txtFTPServerInitialPath.Text
Else
MSettings.AppSettings["FTP/ServerURL"] = ""
MSettings.AppSettings["FTP/Username"] = ""
MSettings.AppSettings["FTP/Password"] = ""
MSettings.AppSettings["FTP/InitialPath"] = ""
Endif
MSettings.AppSettings.Save()
End

```

24.2.3.1.4 Programm – FTPCLIENT

Der Programm-Quelltext wird nur in ausgewählten Abschnitten angegeben und ist dort hinreichend kommentiert. Das komplette Projekt-Archiv finden Sie im Downloadbereich.

Es hat sich während der Programm-Entwicklung als sehr hilfreich erwiesen, die Debug-Eigenschaft

```

'-- OPTION
hFTPClient.Debug = True

```

auf den Wert True zu setzen.

24.2.3.1.5 Quelltext: Anzeige von Verzeichnissen und Dateien auf dem FTP-Server

Die folgenden drei Prozeduren realisieren die Anzeige der relevanten Verzeichnisse und der Datei-Namen (mit passendem Symbol) sowie ausgewählter Datei-Eigenschaften (→ Abbildung 24.2.3.1.1):

```

Public Sub GetServerDirectories()

    Dim n As Integer
    Dim sLine, sSize, sDate, sFileName As String
    Dim asDirRawData As New String[]
    Dim asDirData As New String[]

    asServerDirData = New String[]
    avServerFileData = New Variant[]
    aoFileIcon = New Object[]

    Inc Application.Busy
    hFTPClient.Async = False
    hFTPClient.URL = sFTPServerURL &/ sFTPServerInitialPath &/ txtServerDir.Text &/ "/"

    '-- Reads the contents of the specified server directory and stores it in a temporary file
    hFTPClient.Get(sTempFileName)

    If hFTPClient.ErrorText Then
        Message.Error(("The server reports the error") & ":\n" & hFTPClient.ErrorText)
        '-- The main program is terminated without comment!
        FMain.Close()
    Endif

    asDirRawData = Split(File.Load(sTempFileName), gb.NewLine, "\\", True)

    For Each sLine In asDirRawData
        '-- Several contiguous spaces in the string are reduced to 1 space character
        While InStr(sLine, " ")
            sLine = Replace(sLine, " ", " ")
        Wend

        '-- Last parameter avoid empty items on asDirData
        asDirData = Split(sLine, " ", True)

        '-- . and .. are the convention names of actual and parent Directory we avoid them if exist
        If asDirData[8] = "." Or If asDirData[8] = ".." Then Continue

        sFilename = ""
        '-- Element 8 is only the *first* word of the name. There can also be more ...
        For n = 8 To asDirData.Max
            sFileName &= asDirData[n] & " "
        Next
        '-- Delete the last space added in the last for-next loop.
        sFileName = Left(sFileName, -1)

        If asDirData[0] Begins "d" Then ' ← The rights string for a directory begins with `d`

```

```

'-- Add-Option: asDirData[8], we use the local variable `sFilename` instead
asServerDirData.Add(sFilename)
Else
  sSize = GetFileSize(CLong(asDirData[4])) ' ← File-Size
  sDate = asDirData[6] & " " & asDirData[5] & " " & asDirData[7] ' ← Day - Month - Time
'-- Filename, Filesize and Date of last change
avServerFileData.Add([sFileName, sSize, sDate])

'-- Returns the icon associated with a specific file. (DesktopMime class)
File.Save("/tmp" & "/" & File.Ext(sFileName), "")
aoFileIcon.Add(Desktop.GetFileIcon("/tmp" & "/" & File.Ext(sFileName), 16)) ' ← File-Icon
Endif
Next
Dec Application.Busy
End

Public Sub ShowServerDirectories()

  Dim sServerDirName As String

  For Each sServerDirName In asServerDirData
'-- sServerDirName = Conv$(sServerDirName, "ISO-8859-15", "UTF-8") ' → After a tip from Claus D.
  If trvServerDirectories.MoveTo(txbServerDir.Text & "/" & sServerDirName) Then
    trvServerDirectories.Add(txbServerDir.Text & "/" & sServerDirName, \
      Stock["directory"], txbServerDir.Text, Null).EnsureVisible
  Endif
  Next
End

Public Sub ShowServerFiles()

  Dim iRow, iColumn As Integer
  Dim avLines As New Variant[]

  grvServerFiles.Rows.Count = avServerFileData.Count

  For iRow = 0 To avServerFileData.Max
    avLines = avServerFileData[iRow]
    grvServerFiles[iRow, 0].Picture = aoFileIcon[iRow]
    For iColumn = 0 To avLines.Max
'-- grvServerFiles[iRow, iColumn].Text = Conv$(avLines[iColumn], "ISO-8859-15", "UTF-8")
    grvServerFiles[iRow, iColumn].Text = avLines[iColumn]
    Next
  Next
End

Public Sub trvServerDirectories_Click()

  txbServerDir.Text = trvServerDirectories.Key
  trvServerDirectories[trvServerDirectories.Key].Expanded = IIf(trvServerDirectories.Key = "/", \
  True, Not trvServerDirectories[trvServerDirectories.Key].Expanded)

  txbServerDir_Activate()
End

```

Hinweis: Für die Sortierung der Dateien auf dem FTP-Server gilt: Zuerst Ziffern, dann Großbuchstaben und dann kleine Buchstaben (→ Abbildung 24.2.3.1.1).

24.2.3.1.6 Quelltext: Datei-Download

Ob Sie nur eine Datei oder mehrere markierte Dateien vom FTP-Server in ein markiertes lokales Verzeichnis herunterladen wollen oder den Download mit Drag&Drop realisieren – Sie nutzen stets die folgende Prozedur RunDownload():

```

Private Sub RunDownload()

  Dim sServerFileName, sServerFilePath As String
  Dim aiRows As New Integer[]
  Dim n, iIndex As Integer

  If grvServerFiles.Rows.Selection.Count = 0 Then
    Message.Info(("No file was selected!"))
    Return
  Endif

```

```

pbarDownload.Visible = True
pbarDownload.Value = 0

Inc Application.Busy

aiRows = grvServerFiles.Rows.Selection

For n = 0 To aiRows.Max
    pbarDownload.Value = 0

    iIndex = aiRows[n]
    sServerFileName = grvServerFiles[iIndex, 0].Text
    lblDownloadFilename.Text = " → " & sServerFileName

    hFTPClient.Async = True
    hFTPClient.URL = sFTPServerURL & sFTPServerInitialPath & txbServerDir.Text & sServerFileName
    sServerFilePath = txbLocalDir.Text & sServerFileName

    hFTPClient.Get(sServerFilePath) ' ← File-Download

    While hFTPClient.Status > 0
        If hFTPClient.TotalDownloaded > 0 Then
            pbarDownload.Value = hFTPClient.Downloaded / hFTPClient.TotalDownloaded
        Endif
        Wait 0.01
    Wend

    If hFTPClient.ErrorText Then
        Message("The server reports the error") & ":\n" & hFTPClient.ErrorText
    Else
        pbarDownload.Value = 1
        Wait 0.5
        dirvLocalDirectories.Reload()
    Endif
Next
Dec Application.Busy

```

End

24.2.3.1.7 Quelltext: Datei-Upload

Ob Sie nur eine Datei oder mehrere markierte Dateien aus einem lokales Verzeichnis in ein markiertes Verzeichnis auf dem FTP-Server hochladen wollen oder den Upload mit Drog&Drop realisieren – Sie nutzen stets die folgende Prozedur RunUpload():

```

Private Sub RunUpload()

    Dim sLocalFileName, sLocalFilePath As String

    If filevLocalFiles.Selection.Count = 0 Then
        Message.Info("No file was selected!")
        Return
    Endif

    pbarUpload.Visible = True
    pbarUpload.Value = 0

    Inc Application.Busy
    For Each sLocalFileName In filevLocalFiles.Selection
        pbarUpload.Value = 0

        lblUploadFilename.Text = " → " & sLocalFileName

        '-- Here it is possible to give the file to be uploaded a different name.
        '-- This is unnecessary, however, as files can be renamed on the server.
        hFTPClient.Async = True
        '-- Currently the file name is kept when uploading to the server
        hFTPClient.URL = sFTPServerURL & sFTPServerInitialPath & txbServerDir.Text & sLocalFileName
        sLocalFilePath = filevLocalFiles.Dir & sLocalFileName

        hFTPClient.Put(sLocalFilePath) ' ← File-Upload

        While hFTPClient.Status > 0
            If hFTPClient.TotalUploaded > 0 Then
                pbarUpload.Value = hFTPClient.Uploaded / hFTPClient.TotalUploaded
            Endif
            Wait 0.01
        Wend

        If hFTPClient.ErrorText Then
            Message("The server reports the error") & ":\n" & hFTPClient.ErrorText
        Endif
    Next
Dec Application.Busy

```

```
    Else
        pbarUpload.Value = 1
        Wait 0.5
        RefreshServerView()
    Endif
Next
Dec Application.Busy
End
```