

24.1.4.0 Unix-Domain-Socket (Unix-Protokoll)

Dieses Kapitel basiert auf den Kapiteln 24.1.1 Socket und 24.1.2 ServerSocket.

Beachten Sie, dass Unix-Domain-Sockets genau genommen keine Sockets im Sinne der Netzwerkprogrammierung sind und zum Thema Interprozesskommunikation (IPC) gehören → Kapitel 27.5.0 XML-RPC ( Remote Procedure Call). Der Grund ist verständlich, weil Dateien anstelle von IP-Adressen und Ports zur Kommunikation zwischen den Sockets verwendet werden. Diese Dateien werden im Dateisystem automatisch angelegt, wenn ein Socket an eine Datei gebunden wird. Unix-Domain-Sockets arbeiten verbindungsorientiert, voll duplex und sind schneller als TCP-Sockets.

So erzeugen Sie einen neuen Unix-Server-Socket:

```
Public UNIXServerSocket As ServerSocket
Private sUnixPath As String

sUnixPath = User.Home &/ "socket" ' Beispiel für eine Pfad-Angabe.

Private Sub CreateUNIXServerSocket()

'-- Es wird ein UNIX-Server-Socket erzeugt
UNIXServerSocket = New ServerSocket As "UNIXServerSocket"
'-- Der Server-Typ wird durch die Typ-Angabe auf `Unix-Domain-Socket` festgelegt
UNIXServerSocket.Type = Net.Unix ' Net.Local
UNIXServerSocket.Path = sUnixPath ' An Stelle von ServerSocket.Port

End
```

Hinweise:

- **Synonym** kann für die Konstante *Net.Unix* auch *Net.Local* für verwendet werden, da beide Konstanten den Wert 0 haben.
- Die Angabe 'socket' ist frei wählbar – muss aber für den Unix-Domain-Server-Socket und den Unix-Domain-Client-Socket gleich sein!

Analog gilt für einen Unix-Client-Socket:

```
Public UNIXClient As Socket
Private sUnixPath As String

sUnixPath = User.Home &/ "socket"

Private Sub CreateUNIXClientSocket()

'-- Es wird ein UNIX-Client-Socket erzeugt
UNIXClient = New Socket As "UNIXClient"
'-- Der Client-Typ wird durch die Port-Angabe (!) auf `Unix-Domain-Socket` festgelegt
UNIXClient.Port = Net.Unix ' Net.Local
UNIXClient.Path = sUnixPath ' An Stelle von ClientSocket.Port

End
```

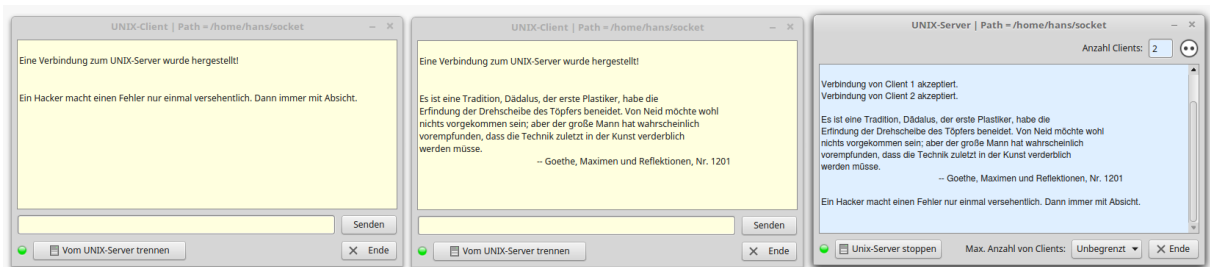


Abbildung 24.1.4.0.1: Unix-Domain-Server-Client-Applikation