

EasyFORM/SQL

**Übernahme von NonStop-SQL Daten
auf PCs**

Datum: 27.04.2000

**Marcus von Cube
B & S GmbH
Borsigstraße 12
65205 Wiesbaden
Tel: 0 61 22 / 77 05-31
Fax: 0 61 22 / 77 05-25
Email: marcus@mvcsys.de**

Inhaltsverzeichnis

1 Einleitung.....	1
2 Installation.....	1
2.1 RemoteSQL auf dem Tandem-System.....	1
2.2 Installation auf dem PC.....	2
2.2.1 PC-Programm EF-SQL.EXE.....	2
2.2.2 MS-Excel-Arbeitsmappe EF-SQL.xls.....	2
3 Einsatz.....	3
3.1 Parameter von EF-SQL.exe.....	3
3.1.1 Parameter auf der Kommandozeile.....	3
3.1.2 Parameter als Steuerdatei.....	4
3.2 Bedienung der Excel-Arbeitsmappe.....	5
3.3 Verwendung von EasyFORM mit anderen Anwendungen.....	7
3.3.1 VBA-Makro aus EF-SQL.xls.....	7
3.3.2 Beispiel einer Ausgabedatei.....	10

1 Einleitung

EasyFORM/SQL ist eine Anwendung der RemoteSQL-Technologie zur Übernahme von Tandem NonStop-SQL-Datenbanken auf PCs. EasyFORM ist ein externes Programm, daß auf dem PC läuft und über das Netzwerk SQL-Befehle an das Tandem-System sendet. Um dem Anwender den Umgang zu erleichtern gehört eine MS-Excel-Arbeitsmappe zum Lieferumfang, die die Funktionen durch einen Dialog zugänglich macht und die Ergebnisse direkt in eine Seite des Arbeitsblattes kopiert.

Für die formatierte Ausgabe von Enscribe-Daten existiert die Anwendung EasyFORM/ENS.

2 Installation

2.1 RemoteSQL auf dem Tandem-System

Um EasyFORM benutzen zu können, ist zuerst *RemoteSQL* auf dem Tandem-System zu installieren:

1. Kopieren Sie RSQLSRV.BIN als Binärdatei auf die Tandem und geben Sie ihr den Filecode 100. Das kann mit folgenden FTP-Befehlen auf dem PC erreicht werden:

```
ftp tandem
(Benutzer und Kennwort)
bin
put RSQLSRV.BIN $SYSTEM.UTIL.RSQLSRV,100
```

2. RSQLSRV muß nun auf der Tandem mit SQLCOMP compiliert werden:

```
SQLCOMP / IN $SYSTEM.UTIL.RSQLSRV / CATALOG $VOLUME.SQLCAT
```

Geben Sie einen beliebigen SQL-Katalog auf ihrem System an.

3. Damit RSQLSRV automatisch gestartet wird, wenn der PC eine Anfrage macht, muß noch die TCP/IP-Konfiguration erweitert werden. Editieren Sie dazu die Datei PORTCONF in ihrem TCP/IP-Konfigurations-Subvolume auf dem Tandem-System. Fügen sei eine Zeile nach folgendem Muster hinzu:

```
741 $SYSTEM.UTIL.RSQLSRV
```

Die Zahl 741 ist die TCP-Portnummer und kann von Ihnen auch anders gewählt werden.

4. Damit die Konfiguration aktiv wird, muß der LISTNER-Prozess neu gestartet werden:

```
STOP $LSN0
LISTNER / NAME $LSN0, PRI 160, TERM $TM01.#A / $SYSTEM.TCPIP.PORTCONF
```

Geben Sie die für Sie zutreffenden Parameter und den Namen Ihrer PORTCONF-Datei an.

Nach erfolgreicher Installation können Sie nun PC-Anwendungen, die *RemoteSQL* benutzen, starten. Geben Sie beim Aufruf dieser Programme bitte die Portnummer, die Sie hier in PORTCONF eingetragen haben, an. Der Default der meisten dieser Anwendungen ist 741.

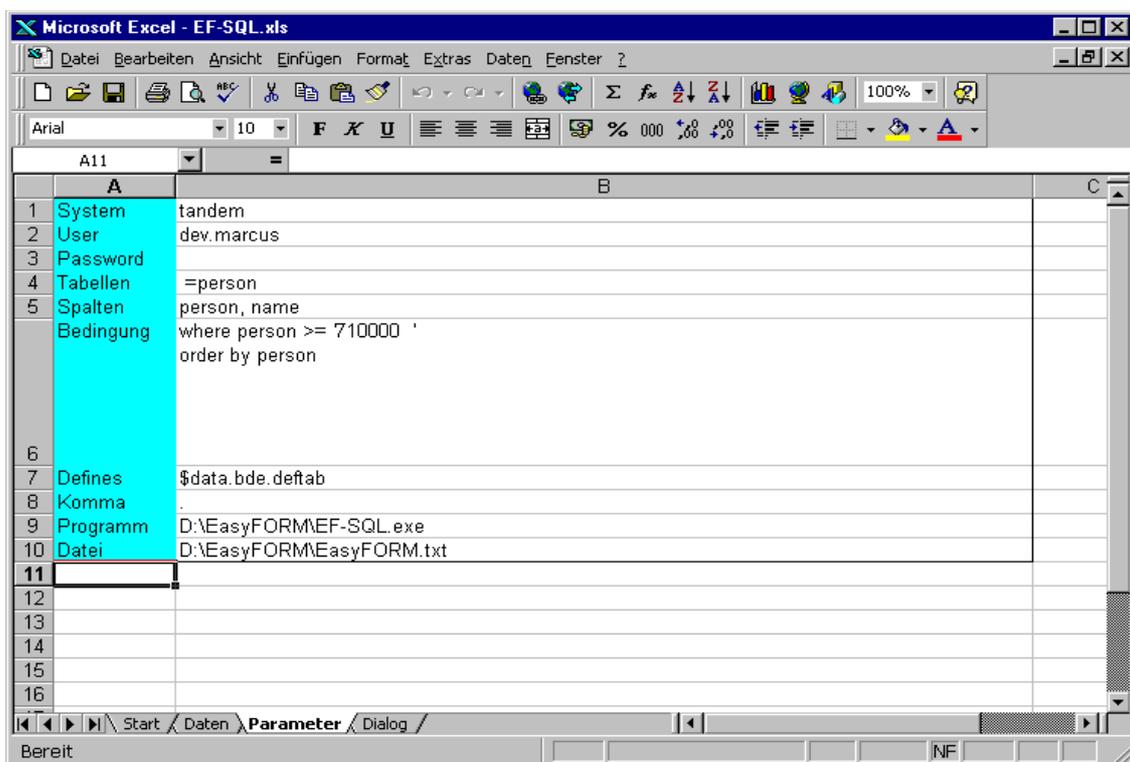
2.2 Installation auf dem PC

2.2.1 PC-Programm EF-SQL.EXE

Kopieren sie EF-SQL.exe in ein beliebiges Verzeichnis auf dem PC oder im Netzwerk. Die aktuelle Version ist als 32-Bit-Programm realisiert. Eine 16-Bit Version, die auch unter Windows 3.x oder mit der 16-Bit-Version von Excel funktioniert, ist auf Anfrage verfügbar¹.

2.2.2 MS-Excel-Arbeitsmappe EF-SQL.xls

Die mitgelieferte MS-Excel-Arbeitsmappe muß noch eingerichtet werden, damit die Vorgaben für Ihre Installation passen. Starten Sie dazu MS-Excel und öffnen Sie die Arbeitmappe. Wechseln Sie auf das Blatt *Parameter*:



Für die Installation sind die Felder *Programm* und *Datei* interessant:

1. Im Feld *Programm* tragen sie die Position von EF-SQL.exe auf Ihrem Rechner ein.
2. Im Feld *Datei* tragen sie den Namen einer Zwischendatei für die Ausgabe der Tandemdaten ein. Die Datei wird von EF-SQL.exe geschrieben und dann vom Excel-Makro in die Arbeitsmappe kopiert.
3. Im Feld *Komma* können sie das Dezimalkommazeichen "." oder "," eintragen. Änderungen sind nur notwendig, wenn der Import von Zahlen nicht korrekt funktioniert, was mit den Einstellungen Ihrer Excel-Version zusammenhängen kann.
4. Die übrigen Felder werden durch den Dialog gesetzt, können aber auch hier vorbelegt werden.

Die geänderte Arbeitsmappe installieren Sie am besten als Vorlage EF-SQL.xlt in Ihrem Excel-Verzeichnis.

¹ Versionen für andere Betriebssysteme wie OS/2 oder Linux, dann allerdings ohne die Unterstützung durch eine Arbeitsmappe, sind auch verfügbar.

3 Einsatz

3.1 Parameter von EF-SQL.exe

EasyFORM.exe muß mit Kommandozeilen-Parametern versorgt werden, damit es arbeitet. Starten Sie das Programm ohne Parameter gibt es eine kurze Anleitung aus:

```
EasyFORM/SQL - Zugriff auf NonStop-SQL - (c) 1997-2000 Marcus von Cube
```

```
EF-SQL <Datei> <Host>[:<port>] <User> "<Pass>" "<Select>" ["<Defines>"]
Alternative: EF-SQL @<Steuerdatei>
```

In der Steuerdatei stehen die Argumente zeilenweise ohne "
Als weitere Parameter koennen folgen:

```
<Dezimalpunkt>  (. oder ,)
<Text-Anfang>   (")
<Text-Ende>     (")
```

3.1.1 Parameter auf der Kommandozeile

Die Parameter können direkt auf der Kommandozeile eingegeben werden. Die Parameter im einzelnen:

1. *Datei* bezeichnet den Namen einer Datei, in der EasyFORM die Ergebnisse der Abfrage speichert. Die Daten werden als Delimited-ASCII erzeugt. Das Trennzeichen ist ein Tabulator. Texte werden in Anführungszeichen eingeschlossen, wenn in den letzten beiden Parametern nichts anderes angegeben wurde. Die erste Zeile der Datei enthält die Spaltennamen. Bei Abfragen über mehrere Tabellen sind die Spaltennamen mit dem Tabellennamen qualifiziert.

TIP: Endet der Name der Zieldatei auf "HTM" oder "HTML" (groß oder klein), dann wird eine Datei im HTML-Format erzeugt, die sich direkt mit einem Internet-Browser anzeigen und ausdrucken läßt.

2. *Host[:Port]* ist die TCP/IP-Adresse des Tandem-Systems. Die Portnummer hinter dem Doppelpunkt kann angegeben werden, wenn sie sich von 741 unterscheidet (siehe Abschnitt 2.1 *RemoteSQL auf dem Tandem-System.*)
3. *User* ist ein gültiger Tandem-Benutzername im Klartext.
4. *Pass* ist das zugehörige Passwort. Wenn das Passwort leer ist, dann geben Sie bitte "" ein.
5. *Select* ist ein beliebiger NonStop-SQL-*SELECT*-Befehl. Dieser kann Define-Namen, Ausdrücke, mehrere Tabellen und sonstige Klauseln für einen gültigen SELECT enthalten. Der Befehl muß in Anführungszeichen eingeschlossen sein, da er Leerzeichen enthält. Schließen Sie Stringkonstanten innerhalb des Befehls bitte in einfache Anführungszeichen ein.
6. *Defines* ist entweder der Name eines Tandem-Obeyfiles mit *ADD DEFINE*-Befehlen oder direkt ein solcher Befehl. Wenn Sie einen Befehl direkt eingeben, dann müssen Sie ihn in Anführungszeichen einschließen.
7. *Dezimalpunkt* ist entweder ein Punkt oder ein Komma und wird benutzt, um skalierte numerische Werte zu formatieren. Default ist ein Punkt.

8. *Text-Anfang* und *Text-Ende* sind Delimiter für Textwerte. Default ist ein doppeltes Anführungszeichen. Es können beliebige -auch leere- Strings angegeben werden. Wird *Text-Ende* nicht angegeben, richtet es sich nach *Text-Anfang*. Sie müssen die beiden Parameter in Anführungszeichen einschließen, wenn sie Sonderzeichen enthalten.

3.1.2 Parameter als Steuerdatei

EasyFORM kann auch mit dem Namen einer Steuerdatei hinter dem Zeichen @ aufgerufen werden. In dieser Datei stehen die Parameter Zeile für Zeile und ohne die begrenzenden Anführungszeichen. Dies ist die Form, die auch das Excel-Makro benutzt. Als Parameterdatei wird vom Makro der gleiche Name wie die Ausgabedatei benutzt. Bricht die Verarbeitung ab, dann enthält die Ausgabedatei also keine Daten sondern die Aufrufparameter!

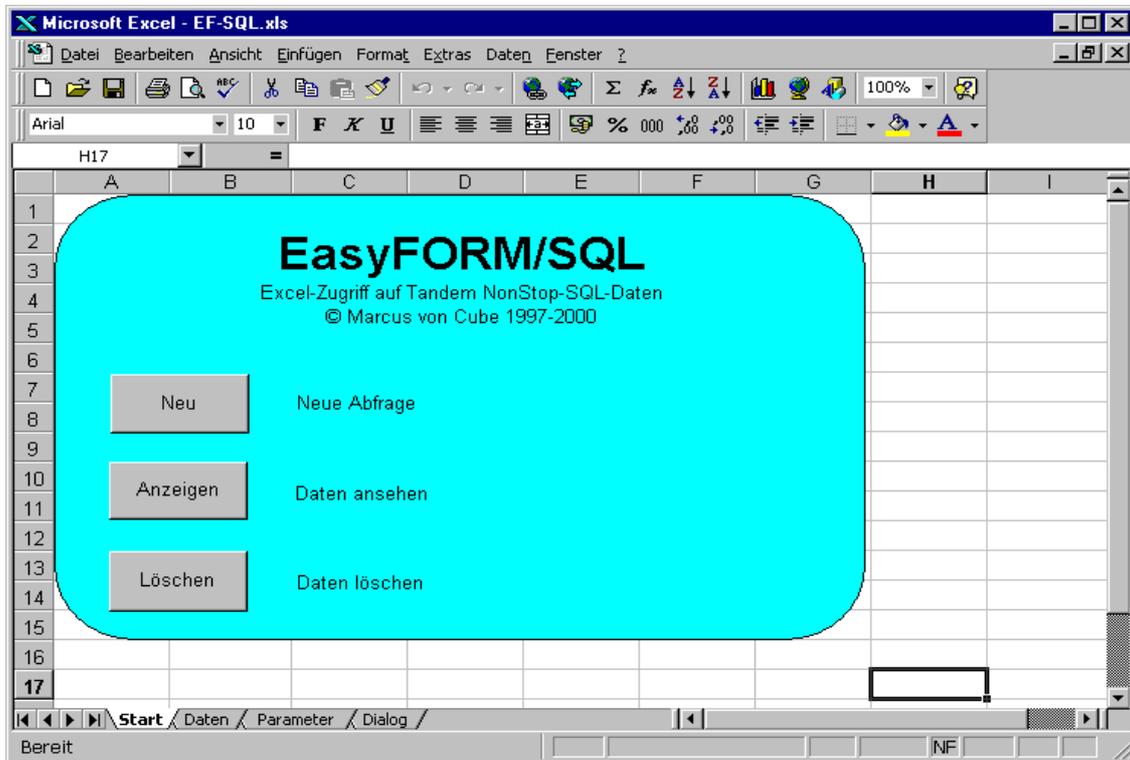
Beispiel einer Steuerdatei:

```
EasyFORM.html
tandem:741
softy.marcus
kennwort
select * from =person where person >= '710000 ' order by person
$easy.bde.defstab
.
"
"
```

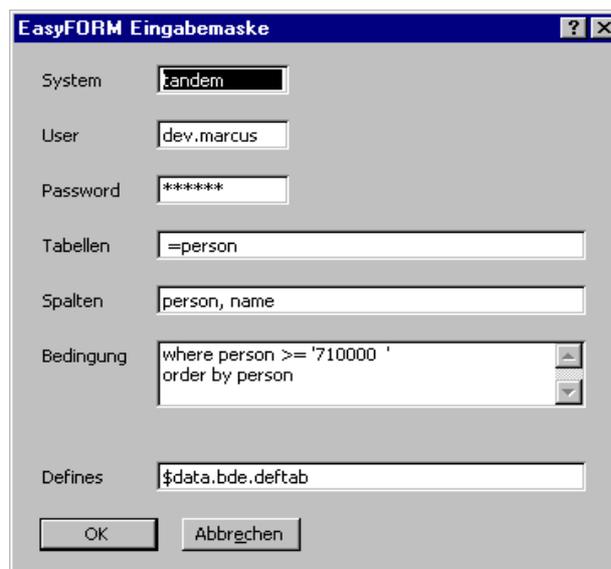
3.2 Bedienung der Excel-Arbeitsmappe

Um EasyFORM über die Excel-Arbeitsmappe zu bedienen, gehen Sie wie folgt vor:

1. Öffnen Sie die Arbeitsmappe oder erstellen Sie aus der Vorlage EasyFORM.xlt eine neue. Wechseln Sie auf das Blatt *Start*:



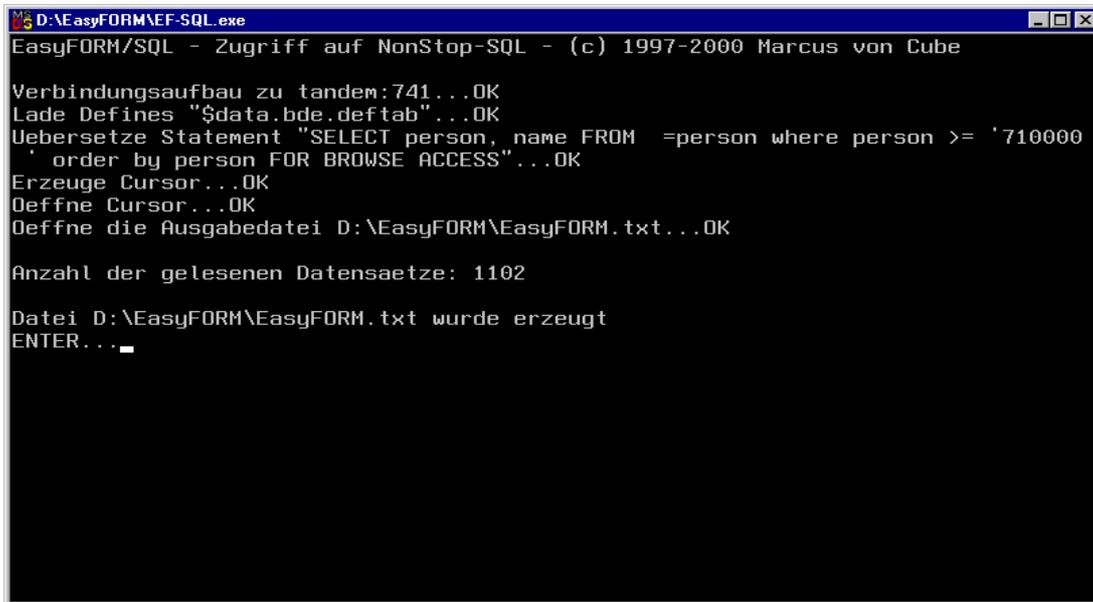
2. Klicken Sie auf *Neu*, um eine Abfrage zu formulieren. Es erscheint der folgende Dialog:



System	tandem
User	dev.marcus
Password	*****
Tabellen	=person
Spalten	person, name
Bedingung	where person >= '710000 ' order by person
Defines	\$data.bde.deftab

3. Füllen Sie die Felder aus. Der Select-Befehl wird aus den Eingaben für die Tabellen, Spalten und die Bedingung automatisch zusammengesetzt.

4. Senden Sie die Dialogbox mit OK ab. Es wird das Programm EasyFORM.exe in einen eigenen Fenster gestartet. Der folgende Bildschirm zeigt die Ausgaben eines erfolgreichen Programmlaufs:



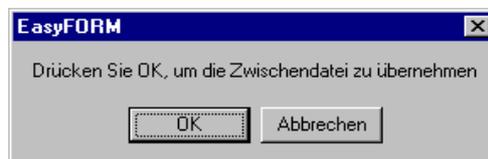
```
D:\EasyFORM\EF-SQL.exe
EasyFORM/SQL - Zugriff auf NonStop-SQL - (c) 1997-2000 Marcus von Cube

Verbindungsaufbau zu tandem:741...OK
Lade Defines "$data.bde.defstab"...OK
Uebersetze Statement "SELECT person, name FROM =person where person >= '710000
' order by person FOR BROWSE ACCESS"...OK
Erzeuge Cursor...OK
Oeffne Cursor...OK
Oeffne die Ausgabedatei D:\EasyFORM\EasyFORM.txt...OK

Anzahl der gelesenen Datensaeetze: 1102

Datei D:\EasyFORM\EasyFORM.txt wurde erzeugt
ENTER..._
```

5. Achten Sie auf Fehlermeldungen, die ebenfalls in diesem Fenster ausgegeben werden. Wenn Die Abfrage fehlerfrei durchgeführt wurde, können Sie das Fenster schließen. Es erscheint wieder Excel mit folgender Abfrage:



6. Wenn EasyFORM korrekt durchgeführt wurde, können Sie den Dialog mit OK bestätigen. Das Excelmakro importiert dann die Zwischendatei in die Seite *Daten* und zeigt diese an. Sie können nun eine neue Abfrage starten oder die gelesenen Daten mit Excel weiter bearbeiten. Speichern Sie ihre Ergebnisse in einer neuen Arbeitsmappe ab. Diese können sie jederzeit wieder benutzen, um die Abfrage zu wiederholen oder zu ändern.
7. Mit der Taste *Anzeigen* des Startbildschirms wechseln Sie auf die Seite *Daten*.
8. Mit der Taste *Löschen* des Startbildschirms löschen Sie den Inhalt der *Daten*-Seite. Dies ist sinnvoll, wenn Sie zwar die Abfrage, nicht aber ihr vollständiges Ergebnis (z. B. wegen der Größe) mit der Arbeitsmappe speichern wollen.

3.3 Verwendung von EasyFORM mit anderen Anwendungen

Andere Anwendungen, zum Beispiel Datenbanken, können EasyFORM ähnlich benutzen wie Excel. Um eine Hilfe für ihrer eigene Makroprogrammierung zu bieten folgt hier der Text des VBA-Makros, das die Excel-Arbeitsmappe steuert.

3.3.1 VBA-Makro aus EF-SQL.xls

```

-----
' EasyFORM
'
' Marcus von Cube 1997-2000
'
' Aufgabe: Zugriff auf Tandem-SQL-Tabellen und Einarbeitung
'          in eine EXCEL-Tabelle
'
' Benoetigt: EasyFORM und Remote-SQL
'
' Datum      Vers.  Bearb.   Erlaeuterung
-----
' 23.06.97  1.0    Mvc      erstellt
'
Option Explicit

Dim Dialog As DialogSheet
Dim Parameter As Worksheet
Dim Daten As Worksheet
Dim OK As Boolean
Dim Datei As String

Sub TandemSQL()
    Set Dialog = DialogSheets("Dialog")
    Set Parameter = Worksheets("Parameter")
    Set Daten = Worksheets("Daten")
    Datei = Parameter.Range("Datei").Value

    Dialog.Buttons("OK").DismissButton = True
    Dialog.Show
    If Not OK Then
        MsgBox "Abgebrochen", 0, "EasyFORM"
        Exit Sub
    End If

    Dim cmd As String
    cmd = Parameter.Range("Programm").Value
    Dim i As Integer
    Dim p As String

    '
    ' Parameter ueber Steuer-Datei
    '
    cmd = cmd + " @" + Datei

    Open Datei For Output As #1
    Print #1, Datei
    Print #1, Parameter.Range("System").Value
    Print #1, Parameter.Range("User").Value
    Print #1, Parameter.Range("Password").Value
    Print #1, "SELECT " + Parameter.Range("Spalten").Value;
    Print #1, " FROM " + Parameter.Range("Tabellen").Value;

```

```
    If Parameter.Range("Bedingung").Value <> "" Then
        p = Parameter.Range("Bedingung").Value
        For i = 1 To Len(p)
            If Mid$(p, i, 1) < " " Then Mid$(p, i, 1) = " "
        Next i
        Print #1, " " + p;
    End If
    Print #1, ""
    Print #1, Parameter.Range("Defines").Value
    Print #1, Parameter.Range("Komma").Value
    Close #1

    'MsgBox cmd
    Shell cmd, 1

    If vbOK = _
        MsgBox("Drücken Sie OK, um die Zwischendatei zu übernehmen", _
            vbOKCancel, "EasyFORM") _
    Then
        Import
    End If

End Sub

'
' Import der Text-Datei
'
Sub Import()
    Dim ImportFile As Workbook
    Daten.Cells.Delete
    Workbooks.OpenText Filename:=Datei, Tab:=True
    Set ImportFile = ActiveWorkbook
    ActiveSheet.Cells.Copy destination:=Daten.Cells
    ImportFile.Close
    Daten.Activate
    'Cells.Select
    Cells.EntireColumn.AutoFit
    ActiveSheet.Cells(2, 1).Select
End Sub

'
' Dialog-Funktionen
'
Sub Dialog_BeiAnzeigen()
    GetParam ("System")
    GetParam ("User")
    GetParam ("Password")
    GetParam ("Tabellen")
    GetParam ("Spalten")
    GetParam ("Bedingung")
    GetParam ("Defines")
    Dialog.Focus = "D_System"
End Sub

Sub Abbrechen_BeiKlick()
    OK = False
End Sub
```

```
Sub OK_BeiKlick()  
  If IsEmpty("System") Then Exit Sub  
  If IsEmpty("User") Then Exit Sub  
  If IsEmpty("Tabellen") Then Exit Sub  
  If Dialog.EditBoxes("D_Spalten").Text = "" Then  
    Dialog.EditBoxes("D_Spalten").Text = "*"
  End If  
  If Left$(Dialog.EditBoxes("D_Tabellen").Text, 1) = "=" Then  
    Dialog.EditBoxes("D_Tabellen").Text = _  
      " " + Dialog.EditBoxes("D_Tabellen").Text  
  End If  
  SetParam ("System")  
  SetParam ("User")  
  SetParam ("Password")  
  SetParam ("Tabellen")  
  SetParam ("Spalten")  
  SetParam ("Bedingung")  
  SetParam ("Defines")  
  OK = True  
End Sub  
  
Function IsEmpty(feld As String) As Boolean  
  Dim d_feld As String  
  d_feld = "D_" + feld  
  With Dialog  
    If .EditBoxes(d_feld).Text = "" Then  
      .Focus = d_feld  
      MsgBox feld + " mu eingegeben werden"  
      .Buttons("OK").DismissButton = False  
      IsEmpty = True  
    Else  
      IsEmpty = False  
    End If  
  End With  
End Function  
  
Sub GetParam(feld As String)  
  Dialog.EditBoxes("D_" + feld).Text = Parameter.Range(feld).Value  
End Sub  
  
Sub SetParam(feld As String)  
  Parameter.Range(feld).Value = Dialog.EditBoxes("D_" + feld).Text  
End Sub  
  
Sub Anzeigen()  
  Worksheets("Daten").Activate  
End Sub  
  
Sub Loeschen()  
  Worksheets("Daten").Cells.Delete  
End Sub
```

3.3.2 Beispiel einer Ausgabedatei

PERSON	NAME	BEARBEITER	UPDATE_ZEIT
"710640"	"Schuchna, Tanja"	"705574"	1998-01-13 15:29:40
"710715"	"Hoefner, Manuela"	"715101"	1998-01-13 15:35:21
"710962"	"Weinzierl, Antigo"	"705574"	1998-01-13 15:32:57
"711028"	"Synde, Gabriele"	"705574"	1998-01-13 15:31:07
"711085"	"Christopf, Belinda"	"711028"	1998-01-13 15:42:23
"711739"	"Thiemig, Maringa"	"711028"	1998-01-13 15:31:55
"713214"	"Roesch, Margarete"	"703595"	1998-01-13 15:28:18
"715101"	"Edenharder, Sabine"	"742876"	1998-01-13 15:43:04
"715821"	"Pirkl, Christa"	"742876"	1998-01-13 15:27:32
"729087"	"Meier, Birgit"	"742876"	1998-01-13 15:22:48
"736421"	"Kajgana, Sadeta"	"742551"	1998-01-13 15:45:03
"738369"	"Catalkava, Fatima"	"742551"	1998-01-13 15:41:34
"740126"	"Zweck, Gerlinde"	"742551"	1998-01-13 15:39:20
"740738"	"Bierschneider, Anne"	"742551"	1998-01-13 15:26:00
"741454"	"Baumer, Gertrud"	"742551"	1998-01-13 15:25:09
"741504"	"Hofbauer, Elisabeth"	"715101"	1998-01-13 15:36:08
"742403"	"Werner, Annemarie"	"715101"	1998-01-13 15:38:04
"742551"	"Nusselt, Rita"	"715101"	1998-01-13 15:26:39
"742876"	"Hofmann, Andrea"	"715101"	1998-01-13 15:36:31
"763847"	"Strauch, Sabine"	"742876"	1998-01-13 15:30:25

Die Spalten sind mit Tabulatoren getrennt. Die Trennzeichen um Zeichenketten sind beim Aufruf von EasyFORM einstellbar (siehe 3.1.1 Parameter auf der Kommandozeile.) Die in diesem Beispiel verwendeten Zahlen sind in der SQL-Tabelle als Strings vereinbart und daher ebenfalls in Anführungszeichen eingeschlossen. Excel kümmert sich darum allerdings nicht und verwandelt sie trotzdem in Zahlen. Das Datumsformat wird von Excel erkannt und richtig interpretiert.

Skalierte Zahlen werden mit dem eingestellten Dezimalkommazeichen, normalerweise ein Punkt, ausgegeben.