

Datenaustausch zwischen SAS und anderen Anwendungen mit DDE und ODBC

Paul Eberhard Rudolph^{*}, Holger Geißler^{**}, Carina Ortseifen^{**}

^{*}Forschungsinstitut für die Biologie landwirtschaftlicher Nutztiere Dummerstorf/Rostock

^{**}Universität Heidelberg, Universitätsrechenzentrum

1. Dynamic Data Exchange (DDE)

1.1 Einführung

Mit dem Begriff **Dynamic Data Exchange** (DDE) wird eine Möglichkeit zum dynamischen Datenaustausch zwischen zwei Windows-Anwendungen bezeichnet. Damit hat man z.B. die Möglichkeit, die in einer SAS-Datendatei bereitgestellten Daten direkt in das Tabellenkalkulationsprogramm Excel zu senden. Umgekehrt kann man aus den Daten in einer Excel-Tabelle sehr leicht eine SAS-Datendatei erzeugen.

Beim DDE ist SAS stets Client, d.h., der dynamische Datenaustausch mit SAS kann (im Unterschied zu ODBC, vgl. Kapitel 2) nur von SAS aus initiiert werden.

Nachdem in Abschnitt 1.2 die Syntax für DDE beschrieben wird, folgen in den weiteren Abschnitten dieses Kapitels einige Beispiele zur praktischen Anwendung von DDE. Der in diesen Beispielen beschriebene Datenaustausch setzt in der Regel die Verwendung von SAS, Version 6.12 und der MS Office 97 Produkte (Excel 97, Access 97 und Word 97) unter dem Betriebssystem Windows 95 oder Windows NT voraus. Das betrifft auch die entsprechenden Beispiele in Kapitel 2. Werden andere Versionen dieser Programme verwendet, dann ergeben sich möglicherweise Abweichungen gegenüber den hier beschriebenen Vorgehensweisen. Den Autoren bekannte derartige Abweichungen werden an der entsprechenden Stelle auch mitgeteilt.

1.2 DDE Syntax

Die DDE Syntax in SAS hat folgendes Aussehen:

```
filename fileref dde 'DDE-triplet' | 'CLIPBOARD' <DDE-options>;
```

Die Teile dieser SAS-Anweisung haben die folgenden Bedeutungen:

<code>fileref</code>	ist ein frei wählbarer Name (analog zur bekannten <code>filename-</code> bzw. <code>libname-</code> Anweisung).
<code>dde</code>	ist für SAS das Schlüsselwort, das anzeigt, daß DDE durchgeführt werden soll.
<code>'DDE-triplet' 'CLIPBOARD'</code>	Es wird entweder ein <code>'DDE-triplet'</code> definiert oder nur <code>'CLIPBOARD'</code> angegeben. Die Syntax für das <code>'DDE-triplet'</code> lautet <code>'Anwendung topic!item'</code> , wobei <i>Anwendung</i> den Namen des Anwendungsprogramms (z.B. Excel), <i>topic</i> den Namen des entsprechenden Dokuments und <i>item</i> die genauen Zeilen- und Spaltenzahlen bezeichnen. Anstatt ein <code>'DDE-triplet'</code> zu definieren und damit eine genaue Anwendung zu benennen, kann man auch einfach auf das <code>'CLIPBOARD'</code> (die Zwischenablage) verweisen.
<code>DDE-options</code>	vgl. SAS-Companion for the MS Windows Environment, Version 6, Second Edition, p. 101

Der einfachste Weg, ein `DDE-triplet` zu definieren, funktioniert folgendermaßen:

1. Man markiert in Excel zunächst den Datenbereich, der übertragen werden soll. (In anderen Programmen funktioniert das ähnlich.) Danach wird dieser Bereich in die Zwischenablage kopiert (*Bearbeiten* → *Kopieren*).
2. Nach dem Wechsel zur aktiven SAS-Sitzung (Windows 3.1 und 3.11: Tastenkombination <Alt> + <Tab>; Windows 95, NT: Anklicken des SAS-Symbols in der Taskleiste) wird die folgende `filename`-Anweisung

```
filename excel dde ' ' ;
```

in den Programmeditor geschrieben.

3. Im SAS-Menü `Options` wird der Punkt `DDE triplet` angeklickt. Daraufhin öffnet sich eine Dialogbox, in der sich das aktuelle `DDE triplet` befindet. Der Text dieses triplets wird in die Zwischenablage kopiert (z.B. mit der Tastenkombination <Strg> + <C>). Anschließend wird der Cursor zwischen die beiden Hochkommata der `filename`-Anweisung plaziert und an diese Stelle der triplet-Text aus der Zwischenablage (z.B. mittels der Tastenkombination <Strg> + <V>) kopiert.

1.3 Datenaustausch von SAS nach Excel

In den folgenden Abschnitten wird anhand von Beispielen der Datentransfer von SAS nach Excel beschrieben.

Bemerkung 1.3.1:

Da SAS als Dezimaltrennzeichen den Punkt verwendet, ist zunächst über die Windows-Systemsteuerung in der Ländereinstellung ein Punkt als Dezimalseparator einzustellen. (Diese Umstellung kann man allerdings durch die Verwendung von Formaten und Informaten, z. B. COMMAXw.d, auch vermeiden.)

Um einen Datenaustausch zwischen SAS und Excel durchzuführen, müssen beide Programme geöffnet sein.

1.3.1 Importieren einer temporären SAS-Datei in Excel

Beispiel 1.3.1:

Es soll eine temporäre SAS-Datei mit dem Namen data131 über eine cards-Anweisung erzeugt und in Excel importiert werden.

Dazu ist folgendes SAS-Programm zu submittieren:

```
/* ❶ Herstellen der SAS-Datei data131 */
data data131;
  input name $   alter gewicht;
  cards;
  Peter 23 67.5
  Anke 34 51.7
  Jochen 56 89.9
  Elke 44 56.3
  ;
run;

/* ❷ DDE filename-Anweisung */
filename test dde `excel | Tabelle1!z1s1:z4s3`;

/* ❸ Senden der Datei zu Excel */
data _null_;
set data131;
file test;
put name alter gewicht;
run;
```

Im Schritt ❶ wird die temporäre SAS-Datei data131 erstellt.

Der Schritt ❷ stellt die Verbindung zur aktiven Excel-Sitzung her und verweist auf die Tabelle1 in die Zeilen 1-4 und die Spalten 1-3.

Der Schritt ❸ vollzieht den eigentlichen Datenaustausch. Dazu wird ein data step ausgeführt, in dem die Datei data131 (set data131) übernommen wird. Danach wird mit einer file-Anweisung diese Datei an die Stelle gesendet, die im Schritt ❷ definiert wurde. Die put-Anweisung bewirkt das Schreiben der entsprechenden Daten. Die Daten werden in die jeweils geöffnete Excel-Datei geschrieben, da im DDE-triplet kein spezifischer Excel-Dateiname angegeben ist. Das Schreiben erfolgt

unabhängig davon, in welcher Tabelle man sich bei Excel befindet oder wo der Cursor steht. Befinden sich an den angegebenen Stellen bereits irgendwelche Daten, werden diese überschrieben.

□

1.3.2 Importieren einer permanenten SAS-Datei in Excel

Beispiel 1.3.2:

Die im Beispiel 1.3.1 erzeugte SAS-Datei `data131.sd2` möge jetzt als permanente Datei im Verzeichnis `C:\SAS\DDEODBC` stehen. Diese Datei soll in Excel importiert werden.

Das entsprechende SAS-Programm unterscheidet sich von dem Programm aus Beispiel 1.3.1 nur dadurch, daß Schritt ❶ wegfällt und dafür eine `libname`-Anweisung notwendig wird, die dem physischen Verzeichnis `C:\SAS\DDEODBC` z.B. den logischen, von SAS verwaltbaren Bibliotheksnamen `dd` zuweist.

```
/* Setzen einer libname-Anweisung */
libname dd 'c:\sas\ddeodbc';

/* ❷ DDE filename-Anweisung */
filename bsp132 dde`excel | Tabelle1!z1s1:z4s3`;

/* ❸ Senden der Datei zu Excel */
data _null_;
    set dd.data131;
    file bsp132;
    put name alter gewicht;
run;
```

Die Daten werden in diesem Beispiel völlig analog zu Beispiel 1.3.1 in die geöffnete Excel-Datei geschrieben.

□

1.3.3 Erstellen einer temporären SAS-Datei mit einem Zufallszahlengenerator und Import in Excel

Beispiel 1.3.3:

Es sollen mit der SAS-Subroutine `ranuni` zehn im Intervall $[0,1]$ gleichverteilte Zufallszahlen erzeugt werden. Der Name der erzeugten Zufallszahlen sei `x`. Außerdem sollen noch die Variablen `y` und `z` erzeugt werden, die sich jeweils aus `x` durch Vergrößerung bzw. Verkleinerung um 10 ergeben. Diese so erzeugten Daten sollen im gleichen `data step` in der `Tabelle1` einer geöffneten Excel-Datei in den Zeilen 1-10 und den Spalten 1-3 positioniert werden.

Das entsprechende SAS-Programm hat dann folgendes Aussehen:

```
filename random dde 'excel | Tabelle1!z1s1:z10s3';
data data133;
  file random;
  do i=1 to 10;
    ranuni(i);
    y=x+10;
    z=x-10;
    put x:6.3 y:6.3 z:6.3;
  output;
  end;
run;
```

□

Die mit dem obigen SAS-Programm erzeugten Daten sind, solange SAS nicht beendet wird, nicht nur in der entsprechenden Excel-Datei sondern aufgrund der output-Anweisung auch in der temporären SAS-Datei data133.sd2 im SAS-Verzeichnis WORK verfügbar.

Da die in SAS und Excel verwendeten Formate häufig nicht übereinstimmen, kann es mitunter zu Problemen hinsichtlich der Darstellung der in Excel übernommenen Daten kommen. Dieses Problem wurde hier durch die Angabe der Formate 6.3 (6 Stellen insgesamt mit 3 Dezimalstellen) für die drei Variablen in der put-Anweisung umgangen. Falls Charakter- oder Datumsvariable von SAS in Excel importiert werden sollen, so funktioniert das mit den entsprechenden Formatangaben (vgl. SAS, Language: Reference, Version 6, First Edition, pp. 673-713).

1.4 Datenaustausch von Excel nach SAS

Excel-Daten können über DDE direkt in SAS eingelesen werden, wenn beide Programme geöffnet sind. In Excel ist die Datei zu öffnen, aus der Daten in SAS übertragen werden sollen. Dabei ist neben der bereits erwähnten Einstellung des Punktes als Dezimalseparator bzw. der Verwendung von Informaten darauf zu achten, daß sich Excel nicht im Editiermodus befindet.

1.4.1 Importieren von Daten aus einer Excel-Datei in SAS

Beispiel 1.4.1:

Es sollen die Daten aus Beispiel 1.3.1, die in der Excel-Datei data131.xls gespeichert wurden, in eine temporäre SAS-Datei mit dem Namen data141 importiert werden. Ein entsprechendes SAS-Programm, das diese Aufgabe löst, hat folgendes Aussehen:

```
filename bsp141 dde
  'excel|c:\sas\ddeodbc\[data131.xls]Tabelle1!Z1S1:Z4S3';
data data141;
  infile bsp141;
  input name $ alter gewicht;
run;
```



Das DDE-triplet im obigen Programm wurde so definiert/kopiert, wie es bereits im Abschnitt 1.2 beschrieben wurde.

Die einfache Datenstruktur in diesem Beispiel dürfte eigentlich keinerlei Fehler verursachen. So ist zum Beispiel die Charaktervariable `name` nicht länger als 8 Zeichen und enthält keine Leerzeichen. Außerdem stehen in den als numerische Variablen `alter` und `gewicht` einzulesenden Excel-Daten tatsächlich nur numerische Werte und keine nichtbesetzten Felder.

Bemerkung 1.4.1:

In der SAS-Version 6.12 führte das Submittieren des obigen SAS-Programms zu keinerlei Fehlern. In der Version 6.11 allerdings wird die entsprechende Excel-Datei nicht gefunden. Dieser Fehler kann durch Reduzierung des DDE-Triplets um die Pfadangabe behoben werden.

Im allgemeinen werden die in SAS zu importierenden Excel-Dateien auch Charakter-Variablen mit mehr als 8 Zeichen oder auch Datums- und Zeitangaben beinhalten. Um diese fehlerfrei in SAS einzulesen, sind folgende Ergänzungen für z.B. eine Charakter-Variable `name` mit 20 Zeichen, eine Datums-Variable `datum` und eine Zeit-Variable `zeit` nötig:

```
infile fileref dlim='09'x notab dsd missover;  
informat name $20;  
input name datum:ddmmyy8. zeit:time.;
```

Diese Ergänzungen haben folgende Bedeutungen:

<code>fileref</code>	steht für die entsprechende Filereferenz (z.B. <code>bs141</code> in Beispiel 1.4.1).
<code>dlim='09'x</code>	bewirkt, daß statt des Leerzeichens das Tabulatorzeichen <code>'09'x</code> (hexadezimale Codierung) als Trennzeichen berücksichtigt wird.
<code>notab</code>	bewirkt, daß SAS die Tabulatorzeichen aus Excel nicht in Blanks verwandelt.
<code>dsd</code>	bewirkt, daß SAS zwei aufeinanderfolgende Tabulatorzeichen als fehlenden Wert (missing value) erkennt.
<code>missover</code>	sorgt dafür, daß SAS nicht automatisch in die nächste Zeile springt, wenn es in einer Zeile keine Daten findet.

Um Datumsangaben richtig einzulesen, werden sie mit dem Informat `ddmmyy8.` belegt. Ein Datumswert kann in der importierten Datei dann in SAS z.B. in Form einer ganzen Zahl erscheinen. Das hängt damit zusammen, daß in SAS die Zeitrechnung am 1.1.1960 mit dem Tag 0 beginnt. Bei Zeitangaben ist dies ähnlich. Ab 00:00:00 werden immer die Sekunden gezählt, z.B. bedeutet 34 die Uhrzeit 00:00:34 Uhr.

Damit Datums- und Zeitangaben wieder in der normalen Schreibweise erscheinen, muß man ihnen innerhalb eines `proc-` oder `data-steps` wieder ein Ausgabeformat zuordnen.

Datums- und Zeitangaben müssen für SAS in einer bestimmten Schreibweise vorliegen, die mit derjenigen in Excel nicht oder nur teilweise übereinstimmt. Sep79 wird SAS z.B. nur dann als Datum akzeptieren, wenn es mit dem richtigen Informat `monyy5.` belegt wird.

Die `Informat`-Anweisung ist nicht DDE-spezifisch, sondern muß immer angegeben werden, wenn formatierte Daten verwendet werden. (Vgl. hierzu auch SAS, Language: Reference, Version 6, First Edition, p. 378 zur `Infile`-Anweisung, pp. 390-392 zum `Informat` und pp. 673-713 zu den verschiedenen Formaten.)

Bemerkung 1.4.2:

Beim Importieren großer Excel-Dateien in SAS ist es äußerst lästig, daß im entsprechenden SAS-Programm die Variablennamen, die sicher schon in der Excel-Datei stehen, erneut angegeben werden müssen. Im folgenden Beispiel wird gezeigt, wie auch die Variablennamen aus einer Exceldatei in eine SAS-Datei übernommen werden können. Die hier vorgestellte programmtechnische Lösung zur Übernahme der Variablennamen stammt von WHITLOCK (1996).

1.4.2 Importieren von Daten aus einer Excel-Datei in SAS mit Übernahme der Variablennamen

Beispiel 1.4.2:

Die Daten der Excel-Datei `data131.xls` aus Beispiel 1.3.1 wurden um eine Zeile nach unten verschoben und in der 1. Zeile die Variablennamen (! Beschränkung auf 8 Zeichen wegen der SAS-Namen-Konventionen) eingetragen. Die Charaktervariable `name` wurde zusätzlich mit einem `$`-Zeichen markiert. Die so modifizierte Excel-Datei sei unter dem Namen `data142.xls` gespeichert. Diese Datei hat dann in den Zeilen 1-5 und den Spalten 1-3 folgendes Aussehen:

name \$	alter	gewicht
Peter	23	76.5
Anke	34	51.7
Jochen	56	89.9
Elke	44	56.3

Um die Excel-Datei `data142.xls` in eine temporäre SAS-Datei mit dem Namen `data142` zu importieren, muß das folgende SAS-Programm (bei geöffneter Excel-Datei) submittiert werden:

```

filename in1 dde 'Excel|Tabelle1!Z1S1:Z1S3 ';
data _null_;
  length n1 - n3 $ 30;
  infile in1 notab dsd dlm='09'x missover;
  input (n1 -n3) ($);
  put (n1 - n3 ) (=);
  array nm (*) n1 - n3;
  call execute
    ( "filename in2 dde 'excel | Tabelle1!z2s1:z5s3';" | |
      "data w; infile in2 notab dsd dlm='09'x missover;"
      input " );
  do i=1 to 3;
  call execute ( nm(i) );
  end;
  call execute ("; run;" );
  stop;
run;

```

□

Bemerkung 1.4.3:

Ist die Anzahl der zu übernehmenden Variablen nicht 3, dann muß das obige Programm entsprechend korrigiert werden. Sollen sehr viele Variable in eine SAS-Datei übernommen werden, dann ist eventuell auch eine Einstellung der Datensatzlänge z.B. über die Option `lrecl=10000` (entsprechend der tatsächlichen Datensatzlänge) jeweils nach der Option `missover` notwendig.

Weiterführende Hinweise dazu, wie man etwa sowohl Variablennamen als auch Variablen-Labels aus einer Excel-Datei in eine SAS-Datei übernehmen kann, findet man unter der WWW-Adresse <http://www-leland.stanford.edu/class/gsb/excel2sas.html>.

1.5 Datenaustausch von MS Word 97 nach SAS

Mitunter hat man bestimmte Daten in einer Word-Tabelle zusammengestellt und möchte diese Daten nun mit bestimmten SAS-Prozeduren weiter verrechnen. Das folgende Beispiel zeigt, wie man die Daten einer Word-Tabelle in eine SAS-Datei importieren kann.

Beispiel 1.5.1

In Word sei eine Tabelle mit folgendem Aussehen erzeugt worden:

Name	Alter	Gewicht
Havemann, Peter	23	67.5
Biebler, Anke	34	51.7
Kummerow, Jochen	56	89.9
Irrgang, Elke	44	56.3

Diese Tabelle wird z.B. unter dem Namen `tabelle.doc` gespeichert. Das ist deshalb wichtig, da der Name der Tabelle in der `filename`-Anweisung angegeben werden muß. Unter der Voraussetzung, daß zunächst nur Word mit der Datei `tabelle.doc` geöffnet ist, sind die folgenden Schritte zum Importieren der Daten aus der obigen Tabelle in SAS mittels DDE durchzuführen:

<p>❶ Die in Word geöffnete Tabelle wird (mit Ausnahme der Kopfzeile) markiert.</p>
<p>❷ Es wird eine Textmarke definiert und diese z.B. <code>tm</code> genannt. (Word 97: <i>Einfügen</i> → <i>Textmarke</i>, Word 6/7: <i>Bearbeiten</i> → <i>Textmarke</i>)</p>
<p>❸ Es wird eine SAS-Sitzung gestartet und das folgende SAS-Programm submittiert:</p> <pre>filename bsp151 dde 'winword tabelle!tm'; data tab; length name \$25; infile bsp151 dlm='09'x notab dsd missover; input name \$ alter gewicht; run;</pre> <p>(Es ist zu beachten, daß in der <code>filename</code>-Anweisung die richtige Textmarkenbezeichnung (<code>tm</code>) und der richtige Name (<code>tabelle</code>) der Tabellendatei stehen.)</p>

Die so erzeugte temporäre SAS-Datei steht im Verzeichnis `Work` von SAS bis zum Beenden der aktuellen SAS-Sitzung. Analog kann eine permanente SAS-Datei erzeugt werden, indem einfach ein zweistufiger Name im Zusammenhang mit einer bestehenden SAS-library angegeben wird.

□

Bemerkung 1.5.1:

Sollte die Tabelle im Beispiel 1.5.1 Bestandteil einer umfangreicheren Word-Datei mit z.B. dem Namen `artikel.doc` sein, dann ist zunächst nur die Tabelle zu kopieren (*Bearbeiten* → *Kopieren*). Danach wird ein neues Word-Dokument geöffnet und in dieses die Tabelle eingefügt. Dieses neue Dokument, das nur die Tabelle enthält, wird jetzt unter dem Namen `tabelle.doc` gespeichert.

Bei Dezimalzahlen ist darauf zu achten, daß diese in der Word-Tabelle mit einem Punkt als Dezimalseparator dargestellt werden. Ist das zunächst nicht der Fall, kann die Tabelle über das Menü *Bearbeiten* → *Suchen/Ersetzen* leicht entsprechend geändert werden.

Werden Datums- oder Zeitangaben in einer in SAS zu importierenden Word-Tabelle verwendet, sind die entsprechenden Hinweise auf Formatangaben in Abschnitt 1.4 zu beachten.

2. Open Data Base Connectivity (ODBC)

2.1 Einführung

ODBC ist ein von Microsoft entwickelter Standard, um Daten zwischen verschiedenen Anwendungen austauschen zu können. ODBC benötigt vier Komponenten:

- **ODBC-fähige Anwendungen** (z.B. SAS mit der Komponente ACCESS TO ODBC, dBase, Excel, Access)
- **ODBC-Datenquellen-Administrator** (befindet sich in der Windows-Systemsteuerung)
- **ODBC-Treiber**
- **Datenquellen (Data Sources)** (Das sind logische Namen für Verzeichnisse (Ordner), in denen sich Dateien einer bestimmten Anwendung befinden, bzw. für bestimmte Dateien.)

Die ODBC-Kommunikation zwischen den verschiedenen Anwendungen findet mit der Datenbanksprache SQL (Structured Query Language) statt.

2.1.1 ODBC-fähige Anwendungen

Da ODBC von Microsoft entwickelt wurde, sind natürlich alle Microsoft Produkte ODBC-fähig, also z.B. Excel und Access. dBase ist ebenfalls eine ODBC-fähige Anwendung. Damit SAS (voll) ODBC-fähig wird, ist die SAS-Komponente SAS/ACCESS to ODBC zu installieren.

2.1.2 ODBC-Datenquellen-Administrator (ODBC-Driver-Manager)

Der ODBC-Datenquellen-Administrator (ODBC-Driver-Manager) ist praktisch die Schaltzentrale zwischen den verschiedenen Anwendungen. Über eine entsprechende Ikone in der Windows-Systemsteuerung kann dieser Manager aufgerufen werden, um z.B.

- die vorhandenen Datenquellen anzuzeigen,
- neue Datenquellennamen zu definieren,
- die installierten ODBC-Treiber anzuzeigen oder
- neue ODBC-Treiber hinzuzufügen.

2.1.3 ODBC-Treiber

Wo genau sich die ODBC-Treiber für die jeweilige Anwendung befinden, muß im Einzelfall überprüft werden. Mit der Installation einer Microsoft-Anwendung wird in der

Regel bereits eine Reihe von ODBC-Treibern in das Windows-Systemverzeichnis kopiert und der ODBC-Datenquellen-Administrator (ODBC-Driver-Manager) in der Windows-Systemsteuerung eingerichtet. Ist MS Office 97 installiert und beim Öffnen des ODBC-Datenquellen-Administrators fehlen benötigte Treiber, dann ist über die Schaltfläche Software in der Systemsteuerung mit der Auswahl *Microsoft Office 97* → *Hinzufügen/Entfernen* das Office-97-setup zu starten. Über die Wahl *Hinzufügen/Entfernen* → *Datenzugriff* → *Option ändern* → *Datenbank-Treiber* → *Option ändern* können bestimmte ODBC-Treiber nachinstalliert werden. (Die analoge Wahl bei Office lautet *Hinzufügen/Entfernen* → *Konverter, Grafikfilter, Datenbankzugriff* → *Option ändern* → *Datenbankzugriff* → *Option ändern*.)

Die folgende Abbildung zeigt den ODBC-Datenquellenadministrator:

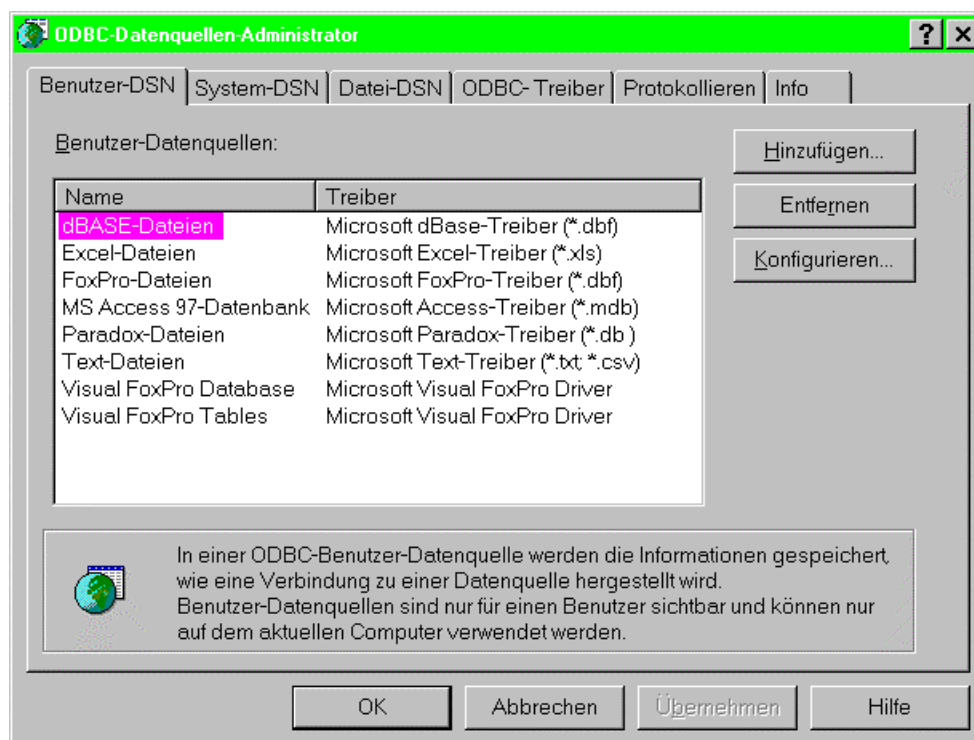


Abb. 2.1: ODBC-Datenquellen-Administrator (Benutzer-DSN)

Die Installation des SAS-ODBC-Treibers wird im Abschnitt 2.2 beschrieben.

2.1.4 Datenquellen

Datenquellen sind logische Namen für Verzeichnisse (Ordner), in denen sich Datendateien einer bestimmten Anwendung (z.B. SAS oder dBase) befinden, oder auch für die Bezeichnung einer Datei mit voller Pfadangabe. Die Definition von Datenquellen erfolgt über ein anwendungsspezifisches setup, das jeweils über die Windows-Systemsteuerung aus dem ODBC-Datenquellen-Administrator heraus aufgerufen werden kann. Die genaue Definition von Datenquellen für bestimmte Anwendungen wird später erläutert.

2.2 SAS-ODBC-Treiber und SAS-Datenquellen

2.2.1 Installation des SAS-ODBC-Treibers

Der SAS-Treiber für ODBC befindet sich auf der Installations-CD oder, falls von einem Netzwerk installiert wird, im entsprechenden Netzwerkverzeichnis. Um ODBC lediglich zum Datenimport in SAS zu verwenden, genügt es, daß SAS/ACCESS to ODBC installiert ist. Falls SAS jedoch auch als Server (Datenexport) verwendet werden soll, muß der ODBC-Treiber für SAS installiert sein.

Um zu prüfen, ob der SAS-ODBC-Treiber bereits installiert ist, wird in der Windows-System-Steuerung die ODBC-Ikone angeklickt, und es öffnet sich das Fenster ODBC-Datenquellen-Administrator. In diesem Fenster sind unter der Rubrik Benutzer-DSN die definierten Benutzer-Datenquellen aufgelistet. Klickt man auf das Feld ODBC-Treiber, dann werden alle installierten ODBC-Treiber aufgelistet.

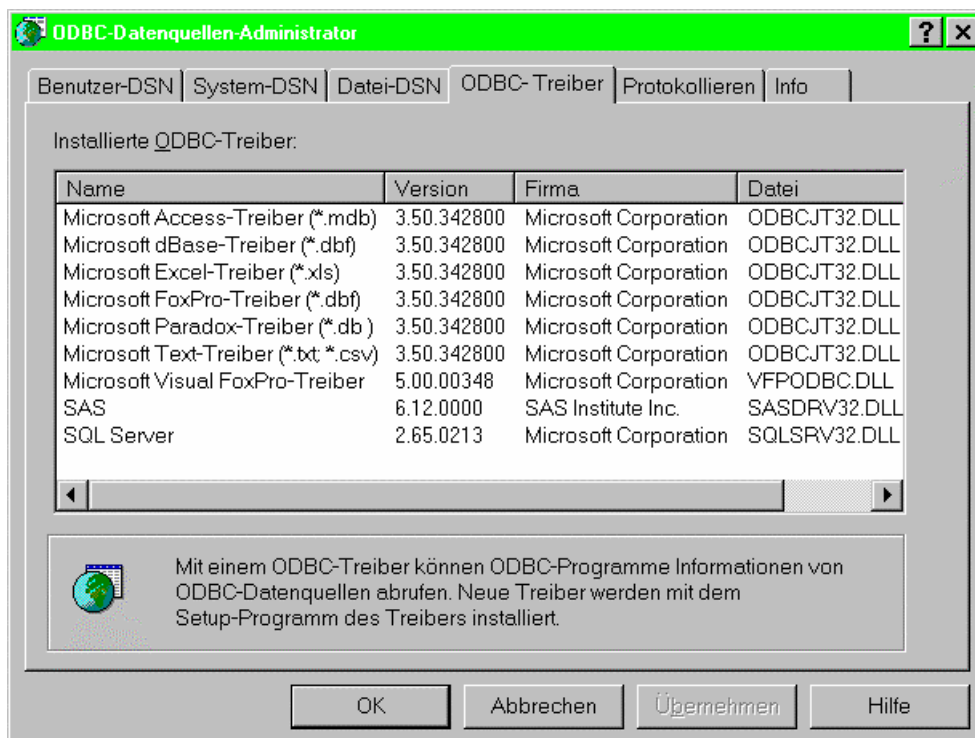


Abb. 2.2: ODBC-Datenquellen-Administrator (ODBC-Treiber)

Ist der SAS-ODBC-Treiber nicht darunter, muß er installiert werden. Im Unterverzeichnis *ODBC* der Installations-CD (oder des SAS-Verzeichnisses auf dem entsprechenden Netzlaufwerk) befinden sich zwei weitere Unterverzeichnisse *setupw16* und *setupw32*. Für Windows 3.1/Windows 3.11 wird der SAS-ODBC-Treiber über *Ausführen* der Datei *drvsetup.exe* im Verzeichnis *seutupw16* und für Windows 95/Windows NT über *Ausführen* der Datei *drvsetup.exe* im Verzeichnis *setupw32* installiert. Nach erfolgreicher Installation öffnet sich der ODBC-Datenquellen-Administrator zur Definition einer SAS-Datenquelle. Diese Definition wird im nächsten Abschnitt beschrieben.

2.2.2 Definition von SAS-Datenquellen

Die Definition einer SAS-Benutzer-Datenquelle kann direkt im Anschluß an die Installation des SAS-ODBC-Treibers erfolgen (vgl. Abschnitt 2.2.1) oder über die Wahl von *Systemsteuerung* → *ODBC* → *Hinzufügen* → *SAS* → *Fertig stellen* . Es wird ein Fenster mit der Überschrift SAS ODBC Driver Configuration geöffnet mit den vier Schaltflächen *General*, *Servers Libraries* und *SQL Options*.

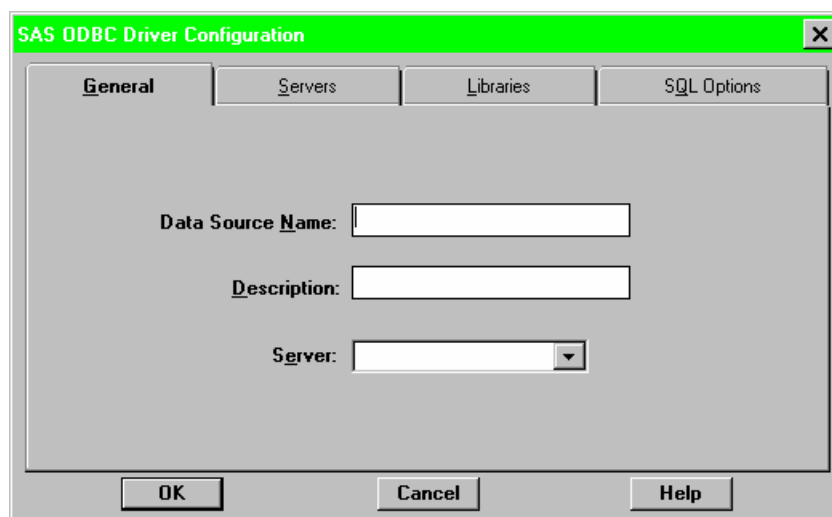


Abb. 2.3: SAS ODBC Driver Configuration (General)

Hier sind zur Fertigstellung der SAS-Datenquellendefinition einige Angaben zu machen. Da einige dieser Angaben optional sind, werden im folgenden nur die notwendigen Angaben aufgeführt.

<i>Data Source Name</i>	Ist ein frei wählbarer Name, ähnlich einer <i>libref</i> .
<i>Server Name</i>	Hier tauchen unter der Schaltfläche <i>General</i> die definierten Server auf. Sollte noch kein Server vereinbart sein, muß das über die Schaltfläche <i>Server</i> geschehen. Als Servername ist stets ein einstufiger Name zu wählen, da damit dem ODBC-Treiber von SAS mitgeteilt wird, daß der Zugriff lokal sein soll. (Zweistellige Server-Namen weisen auf einen Remote-Server (Rechner in einem Netzwerk) hin. Um mit einem solchen Server einen Datenaustausch durchzuführen, wird SAS/SHARE benötigt.
<i>Access Method</i>	Unter der Schaltfläche <i>Server</i> ist bei einer Server-Definition stets <i>DDE</i> für einen lokalen Zugang und <i>Single User Local Server</i> zu wählen (vgl. <i>Server Name</i>). Anschließend ist über die Schaltfläche <i>Config...</i> eine eventuelle Änderung voreingestellter Werte vorzunehmen.
<i>Library Name</i>	Hier muß über die Schaltfläche <i>Libraries</i> ein Name entsprechend einer <i>libname</i> -Anweisung angegeben werden.
<i>Host File Name</i>	Ist die entsprechende Pfadangabe für ein existierendes Unterverzeichnis, dem der angegebene <i>Library Name</i> zugewiesen wird.

Nach dem Fertigstellen der Datenquellendefinition ist die ODBC-Verbindung zu SAS etabliert, und es kann mit dem Datenaustausch zu anderen Programmen begonnen werden.

Bemerkung 2.2.1:

Werden über `libname`-Anweisungen in einer `autoexec.sas` z.B. von SAS automatisch `libraries` verwaltet, dann können die Angaben *Library Name* und *Host File Name* bei der Datenquellen-Definition auch entfallen. Der SAS-ODBC-Server stellt dann automatisch alle SAS-Dateien aus allen verwalteten `libraries` zur Verfügung.

2.3 Definition anderer ODBC Verbindungen

Analog zu den für SAS beschriebenen Schritten lassen sich für andere Anwendungen ebenfalls ODBC-Verbindungen definieren. Das Prinzip ist meist ähnlich, auch wenn der entsprechende Dialog etwas unterschiedlich ist.

2.3.1 Definieren von MS Access-Datenquellen

Nach Wahl von *Systemsteuerung* → *ODBC* → *Hinzufügen* → *Microsoft Access-Treiber (*.mdb)* → *Fertig stellen* öffnet sich ein Fenster mit der Überschrift ODBC Microsoft Access 97-Setup.

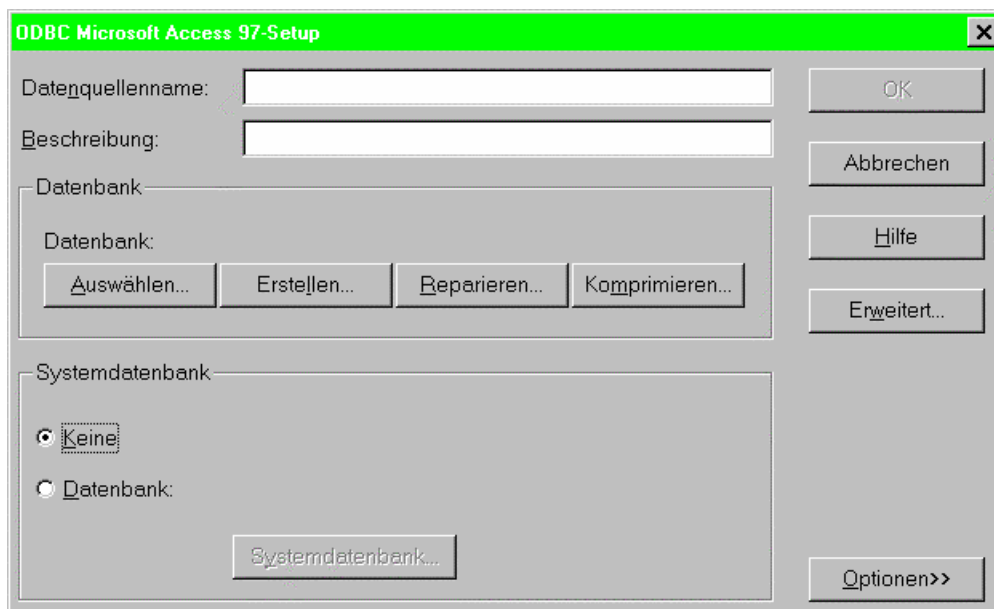


Abb. 2.4: ODBC Microsoft Access 97 - Setup

In diesem Fenster muß dann eine MS Access-Datenbank ausgewählt werden, für die noch ein frei wählbarer logischer *Datenquellenname* anzugeben ist.

2.3.2 Definieren von MS Excel-Datenquellen

Völlig analog zur Definition von MS Access-Datenquellen öffnet sich nach der Wahl von *Systemsteuerung* → *ODBC* → *Hinzufügen* → *Microsoft Excel-Treiber (*.xls)* → *Fertig stellen* ein Fenster mit der Überschrift ODBC Microsoft Excel-Setup .

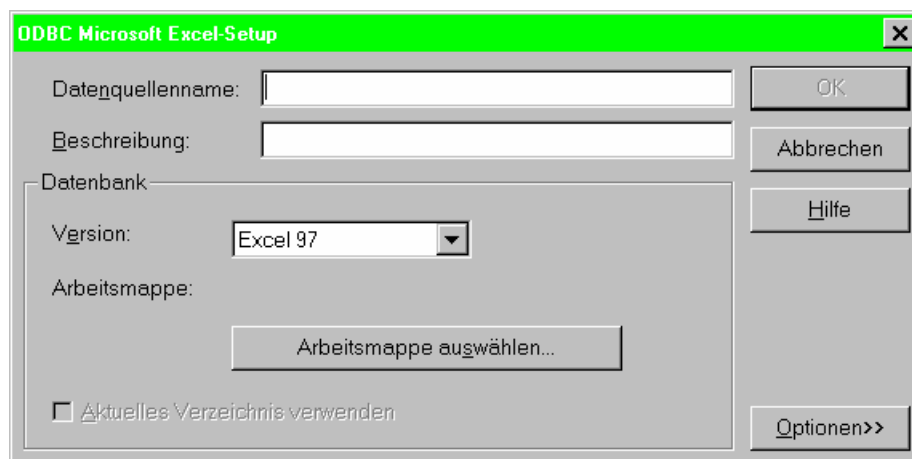


Abb. 2.5: ODBC Microsoft Excel-Setup

Hier ist ebenfalls ein frei wählbarer *Datenquellenname* als logischer Name für eine über die Schaltfläche *Arbeitsmappe auswählen* zu bestimmende Exceldatei anzugeben.

2.3.3 Definieren von dBase-Datenquellen

dBase-Datenquellen lassen sich über die Wahl *Systemsteuerung* → *ODBC* → *Hinzufügen* → *Microsoft Dbase-Treiber (*.dbf)* → *Fertig stellen* definieren.

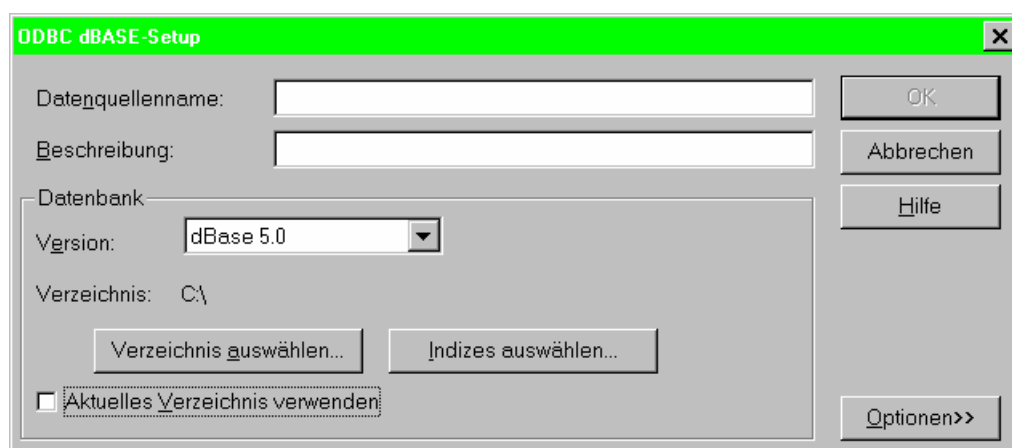


Abb. 2.6: ODBC dBase – Setup

In diesem Fenster (ODBC dBase-Setup) sind dann noch folgende Parameter anzugeben:

<i>Datenquellennamen</i>	Hier wird ein frei wählbarer logischer Name angegeben für ein unter <i>Verzeichnis</i> gewähltes physisches Verzeichnis.
<i>Version</i>	Die verwendete dBase-Version ist auszuwählen.
<i>Verzeichnis</i>	Es kann entweder das aktuelle Verzeichnis verwendet werden oder ein beliebiges anderes. In dem hier anzugebenden Verzeichnis müssen die dBase-Dateien stehen, mit denen kommuniziert werden soll.

Bemerkung 2.3.1:

Nach Auskunft des Technical Support von SPSS in München wird nicht beabsichtigt, einen ODBC-Treiber für SPSS in absehbarer Zeit zur Verfügung zu stellen. SPSS-Datenquellen können deshalb noch nicht definiert werden. SPSS kann deshalb bislang nur Daten anderer Anwendungen über ODBC importieren; andere Anwendungen können allerdings keine SPSS-Daten über ODBC importieren.

2.4 Export von SAS-Daten zu MS Access, MS Excel und SPSS

In den folgenden Abschnitten wird der Export einer SAS-Datendatei zu MS Access, MS Excel und SPSS mit zum Teil konkreten Beispielen demonstriert. Dieser Export ist im Prinzip ein Import, da der Dateitransfer nicht von SAS, sondern aus der jeweils geöffneten Applikation, gestartet wird

2.4.1 Export einer SAS-Datendatei zu MS Access**Beispiel 2.4.1:**

Es soll die SAS-Datei `dd.data131` im Verzeichnis `c:\sas\ddeodbc` in eine Access-Tabelle importiert werden.

Dabei soll vorausgesetzt werden, daß eine SAS-Benutzer-Datenquelle mit dem Datenquellen-Namen `sas1` ohne die Angaben für *Library Name* und *Host File Name* im ODBC-Datenquellen-Administrator definiert wurde. Außerdem soll SAS über die `libname`-Anweisung `libname dd 'c:\sas\ddeodbc';` in einer `autoexec.sas` automatisch die so definierte `library dd` verwalten.

Die im folgenden aufgelisteten Schritte, in denen Fenster-Titel durch Unterstreichen und die Auswahl aus in diesen Fenstern vorhandenen Schaltflächen oder Menüs durch Kursivschrift gekennzeichnet sind, zeigen, wie der gewünschte Dateitransfer zu realisieren ist.

1. Starten von MS Access
2. Erstellen einer leeren oder Öffnen einer bestehenden Access-Datenbank (z.B. `data131.mdb`) → data131: Datenbank .
3. data131: Datenbank: *Tabelle* → *Neu* → Neue Tabelle
4. Neue Tabelle: *Tabelle importieren* → OK → Importieren

5. Importieren: Dateityp: *ODBC-Datenbanken()* → Datenquelle auswählen
6. Datenquelle auswählen: *Datei-Datenquelle* → *Suchen in:* → *(c:\sas\)\ddeodbc* → *Computer-Datenquelle* → *sas1* → *OK* → (SAS startet) → Objekte importieren
7. Objekte importieren: *DD.SAS131* → *OK*
8. Beenden der ODBC-Verbindung in SAS mit der Tastenkombination <Strg> + <Pause> und beenden der SAS-Sitzung mit <Alt> + <F4>

Nach Abarbeitung dieser 8 Schritte wird im Fenster data131: Datenbank neben eventuell bereits existierenden Tabellenbezeichnungen die aus der Datei `data131.sd2` neu erzeugte Tabelle mit dem Namen `DD_DATA131` angezeigt.

□

2.4.2 Export einer SAS-Datendatei zu MS Excel (Version 5.0)

Folgende Schritte sind notwendig, um SAS-Daten mittels ODBC in MS Excel zu importieren:

1. *Daten* → *Daten importieren*
2. Daraufhin wird das MS-Query gestartet. Hier erscheint als erstes erneut eine Liste der definierten Datenquellen. Falls die gewünschte ODBC-Verbindung in der Liste fehlt, bekommt man über die Schaltfläche *Andere* alle im ODBC-Driver-Manager definierten Verbindungen.
3. Daraufhin wird im Hintergrund eine SAS-Sitzung gestartet und mittels DDE werden die Daten übertragen.
4. Man beendet MS Query mit dem Befehl: *Datei* → *Daten an MS Excel zurückgeben*

Bemerkung 2.4.1: Benutzerdefinierte Formate werden nicht übertragen.

2.4.3 Export einer SAS-Datendatei zu SPSS (Version 7.5 für Windows 95)

Der Export einer SAS-Datei zu SPSS funktioniert relativ reibungslos. Folgende Schritte sind dazu nötig:

1. Starten von SPSS Version 7.5
2. Im Neudaten-Fenster (so heißt das leere SPSS-Spreadsheet): *Datei* → *Datenbank einlesen* wählen.
3. Jetzt wird der Assistent für Datenbankzugriff geöffnet (s. Abb. 2.7), in dem alle definierten ODBC-Datenquellen erscheinen. Die SAS-Datenquelle kann hier, falls zuvor definiert, ausgewählt werden.
4. Im Hintergrund wird bei Anwählen der SAS-Datenquelle SAS gestartet.

5. Es erscheint nun eine Liste aller SAS-Dateien, die sich in dem Verzeichnis befinden, auf das die SAS-Datenquelle weist.
6. Um eine SAS-Datei auszuwählen, muß sie mit der Maus in das Auswahlfenster gezogen werden.
7. Als nächsten Schritt kann man entweder den Transfer fertigstellen (*Fertigstellen*) oder sich die SPSS-Syntax für den ODBC-Dialog in das SPSS-Syntax-Fenster ausgeben lassen (dies ist besonders bei wiederholter Anwendung sinnvoll).
8. Beenden der ODBC-Verbindung in SAS mit der Tastenkombination <Strg> + <Pause> und beenden der SAS-Sitzung mit <Alt> + <F4>

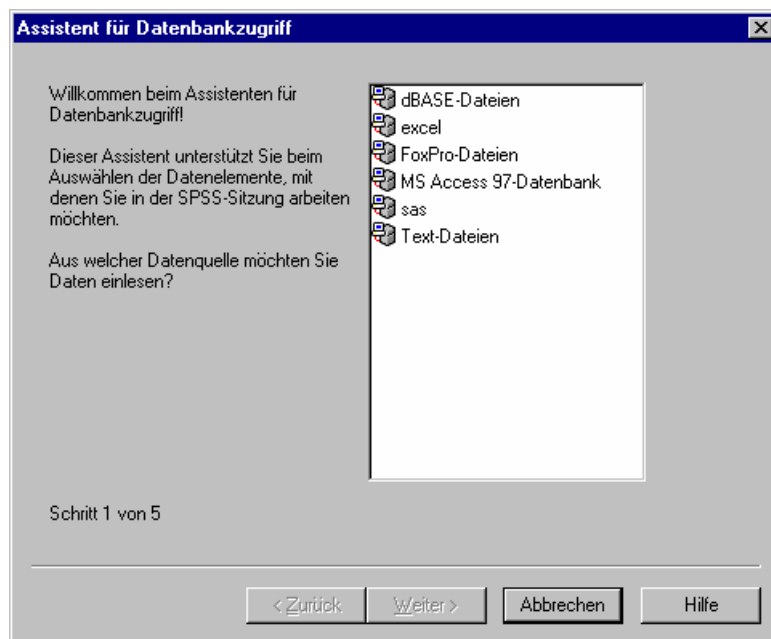


Abb. 2.7: Assistent für Datenbankzugriff

Die Aufschrift 'Schritt 1 von 5' in Abb. 2.7 bezieht sich nicht auf die Numerierung der oben beschriebenen Schritte, sondern ist eine SPSS-spezifische Angabe.

Bemerkung 2.4.2:

Beim Import von SAS-Daten in SPSS werden nur die Variablennamen übernommen, nicht aber die Variablenlabels. Außerdem werden keine Zahlenformate übernommen, jede Variable bekommt das Standardformat 8.2 zugewiesen. Die Charakterformate werden dagegen übernommen.

Eine Möglichkeit diese Probleme zu umgehen, liegt darin, sich die SPSS-Syntax des ODBC-Dialogs ausgeben zu lassen und entsprechend die Formate und Labels hinzuzufügen. Dies setzt allerdings Kenntnis der SPSS-Syntax voraus.

2.5 Import von Daten in SAS aus MS Access und dBase

Nachdem die gewünschte ODBC-Verbindung definiert wurde (vgl. Abschnitt 2.3), greift man innerhalb von SAS mittels der Prozedur SQL auf diese Verbindung zu. Im Gegensatz zum Export von SAS-Daten wird die Anwendung, aus der die zu importierenden Daten stammen, nicht gestartet.

2.5.1 Import einer MS Access-Datei

Beispiel 2.5.1:

Es sollen die Daten aus der Tabelle1 einer Access-Datei mit dem Namen `data131.mdb`, die mit dem ODBC-Datenquellen-Namen `access` verknüpft ist (vgl. Abschnitt 2.3.1), in eine permanente SAS-Datei mit dem Namen `dd.data251` importiert werden.

Ein SAS-Programm, das diesen Daten-Import realisiert, kann in drei Schritte unterteilt werden. Es ist in der folgenden Tabelle dargestellt.

```
/* ❶ ODBC-Verbindung mit der Access-Datenquelle access über proc sql */
proc sql;
connect to odbc (dsn=access);

/* ❷ Erstellen der SAS-Datei data251 */
create table dd.data251 as select * from connection to odbc
(select * from [Tabelle1]);

/* ❸ Schließen der ODBC-Verbindung und beenden von proc sql */
disconnect from odbc;
quit;
```

□

Bemerkung 2.5.1:

Im Vergleich zu DDE übernimmt SAS deutlich mehr Standardformate von der Datenquelle. Auftretende Probleme mit bestimmten Formaten beim Datenaustausch bekommt man in der Regel durch die Verwendung von Informaten und Formaten in den Griff.

2.5.2 Import einer dBase-Datei

Beispiel 2.5.2:

Es sollen die Daten aus einer dBase-4-Datei mit dem Namen `data131.dbf`, die in einem Verzeichnis steht, das mit dem ODBC-Datenquellennamen `dbase4` verknüpft ist (vgl. Abschnitt 2.3.3), in eine permanente SAS-Datei mit dem Namen `dd.data252` importiert werden.

Mit `dd` ist hier die über die `libname`-Anweisung

```
libname dd 'c:\sas\ddeodbc';
```

definierte `library` bezeichnet.

Analog zum Beispiel 2.5.1 kann ein entsprechendes SAS-Programm in die folgenden drei Schritte unterteilt werden:

```
/* ❶ ODBC-Verbindung mit der dBase-Datenquelle dbase4 über proc sql */
proc sql;
connect to odbc (dsn='dbase4');

/* ❷ Erstellen der SAS-Datei data252 */
create table dd.data252 as select * from connection to odbc
(select * from data131.dbf);

/* ❸ Schließen der ODBC-Verbindung und beenden von proc sql */
disconnect from odbc;
quit;
```

□

Literatur

Geißler, H.; Ortseifen, C.: Datenaustausch mit SAS. Universitätsrechenzentrum Heidelberg, 1996. 20 S.

SAS Institute Inc.: SAS Language: Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc., 1990. 1042 pp.

SAS Institute Inc.: SAS ODBC Driver Technical Report: Users Guide and Programmer's Reference, Release 6.11, Cary, NC: SAS Institute Inc., 1995. 50 pp.

SAS Institute Inc.: SAS Companion for the Microsoft Windows Environment, Version 6, Second Edition, Cary, NC: SAS Institute Inc., 1996. 302 pp.

Rudolph, P. E.: Datenaustausch mit SAS über DDE und ODBC. Forschungsinstitut für die Biologie landwirtschaftlicher Nutztiere, Dummerstorf/Rostock 1997

Whitlock, I.: Mail with Subjekt: 'Can SAS DDE to Excel and get Field names in First Row?' posted on SAS-L email list / comp.soft-sys.sas newsgroup, 27 June 1996

Adressen der Autoren:

Dr. Paul Eberhard Rudolph
Forschungsinstitut für die Biologie landwirtschaftlicher Nutztiere
Forschungsbereich Biometrie
Wilhelm-Stahl-Allee 2
18196 Dummerstorf
Tel. : 038208/68-908
Fax : 038208/68-602
e-mail : rudolph@fhn.uni-rostock.de
www : <http://www.fhn.uni-rostock.de/fb8/rudolph/deutsch/hppedeu.htm>

Holger Geißler
Universität Heidelberg, Universitätsrechenzentrum
In Neuenheimer Feld 293
69120 Heidelberg
Tel. : 06221/54-4531
Fax : 06221/54-5581
e-mail : holger.geissler@urz.uni-heidelberg.de

Dr. Carina Ortseifen
Universität Heidelberg, Universitätsrechenzentrum
In Neuenheimer Feld 293
69120 Heidelberg
Tel. : 06221/54-4541
Fax : 06221/54-5581
e-mail : carina.ortseifen@urz.uni-heidelberg.de
www : <http://www.rzuser.uni-heidelberg.de/~x16/>