

Allgemeines zu WrkDbFil

WrkDbFil ermöglicht verschiedene Hilfsfunktionen mit dBase III Dateien.

Im Programm wird eine andere Möglichkeit eines Menüs zur Funktionsauswahl angewendet.

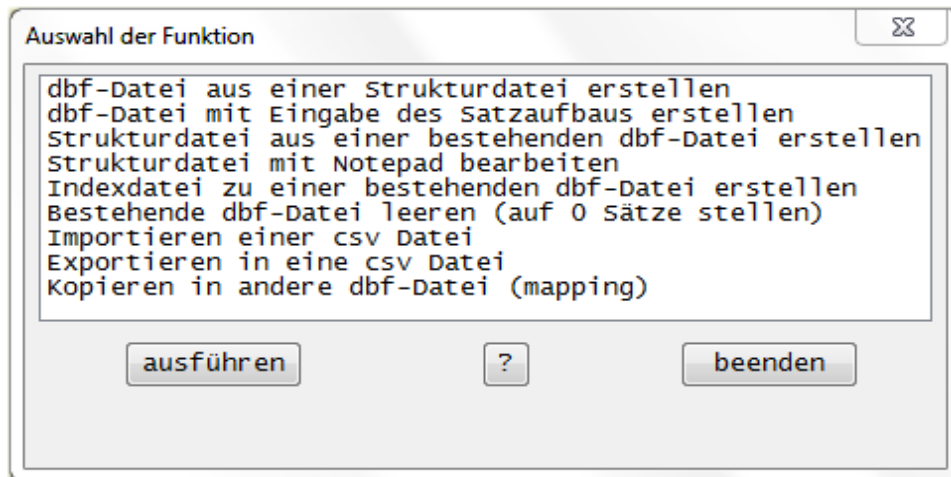


Bild1.

Durch markieren (anklicken) einer Auswahl und Betätigung des **ausführen** Buttons wird die Auswahl (= Funktion) gestartet.

Durch anklicken einer Auswahl und Betätigung des **?** Buttons wird direkt die Hilfe zu dieser Funktion angezeigt.

Das Programm darf frei benutzt werden. Es wurde eingehend getestet, aber es kann keine Garantie übernommen werden falls aus irgendeinem Grund ein Fehler in der Datei produziert wird.

Die in dieser Hilfe gezeigten Abbildungen können anders aussehen, da die Dialoge, abhängig von der Windowsversion, unterschiedlich angezeigt werden.

Autor: Gerhard Putschalka

H.P.: <http://www.gerhard-putschalka.xprofan.com/index.htm>

[weiter zur Funktionsübersicht](#)

Übersicht über die Funktionen

- [dbf-Datei aus einer Strukturdatei erstellen](#)
 - [dbf-Datei mit Eingabe des Satzaufbaues erstellen](#)
 - [Strukturdatei aus einer bestehenden dbf-Datei erstellen](#)
 - [Strukturdatei mit Notepad bearbeiten](#)
 - [Indexdatei zu einer bestehenden dbf-Datei erstellen](#)
 - [Bestehende dbf-Datei leeren \(auf 0 Sätze stellen\)](#)
 - [Importieren einer CSV Datei](#)
 - [Exportieren in eine CSV Datei](#)
 - [Kopieren in eine andere dbf-Datei \(Mapping\)](#)
-
- [Was ist eine CSV-Datei?](#)
 - [Was ist eine Strukturdatei?](#)

[Anhang A.](#)

[zurück](#)

dbf-Datei aus einer Strukturdatei erstellen

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der [Strukturdatei](#)

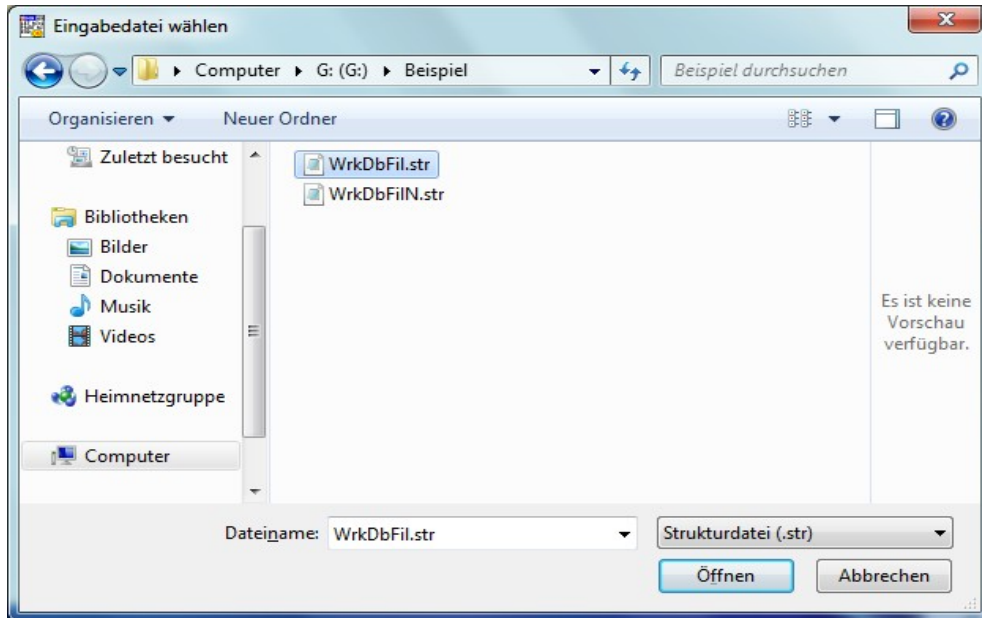


Bild2.

Nachdem die Struktur Datei ausgewählt wurde, wird der Dialog zur Auswahl/Eingabe des Namens für die zu erstellende dBase Datei angezeigt

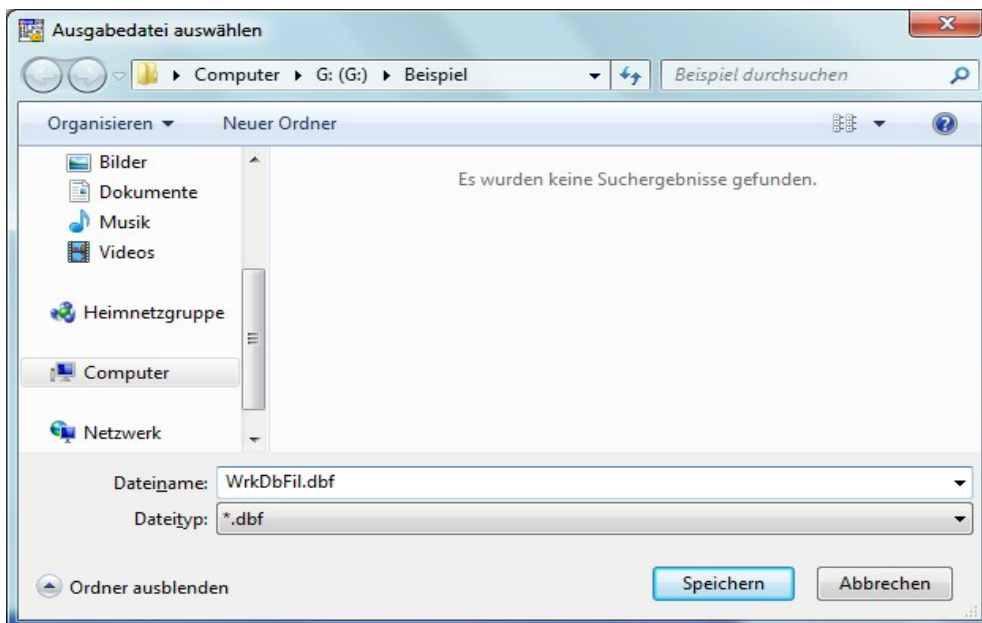


Bild3.

Wird eine Datei ausgewählt, die bereits existiert, erscheint eine Warnung und es besteht die Möglichkeit hier den Vorgang abzubrechen.

[zurück zur Funktionsübersicht](#)

dbf-Datei mit Eingabe des Satzaufbaues erstellen

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Eingabe der Feldbeschreibungen.

Dialog 'Datei definieren' zur Eingabe der Feldbeschreibungen:

- Feldname: ALFA1
- Feldart:
 - ☒ Zeichen
 - ☐ numerisch
 - ☐ Datum
 - ☐ logisch
 - ☐ Memo
- Feldlänge: 15
- Anz. Dezimalstellen: 0
- Dateneingabe mit <ENTER> bestätigen
- Buttons: Datei erstellen, Abbruch, ?, Eintrag entfernen

Bild4.

Als Mindesteingaben sind der Feldname einzugeben und die Feldart zu markieren. Nur bei Feldart **Zeichen** und **numerisch** muss noch die Feldlänge und bei **numerisch** auch noch die Anzahl der Dezimalstellen eingegeben werden. Ein Zeichenfeld kann bis zu 254 Stellen und ein numerisches Feld bis zu 19 Stellen lang sein. Die Anzahl der Dezimalstellen kann 0 bis Feldlänge minus 2, jedoch höchstens 15 Stellen lang sein. Jede Felddefinition ist mit der **Enter**taste zu bestätigen.

Dialog 'Datei definieren' zur Eingabe der Feldbeschreibungen:

- Feldname:
- Feldart:
 - ☐ Zeichen
 - ☐ numerisch
 - ☐ Datum
 - ☐ logisch
 - ☒ Memo
- Feldlänge: 10
- Anz. Dezimalstellen: 0
- Dateneingabe mit <ENTER> bestätigen
- Buttons: Datei erstellen, Abbruch, ?, Eintrag entfernen

ALFA1	C	015	00
NUM120	N	012	00
NUM122	N	012	02
DAT	D	008	00
LOG	L	001	00
MEM	M	010	00

Bild5.

So soll in diesem Beispiel der Satzaufbau der zu erstellenden Datei aussehen.

Mit **Datei erstellen** wird der nächste Dialog angezeigt.

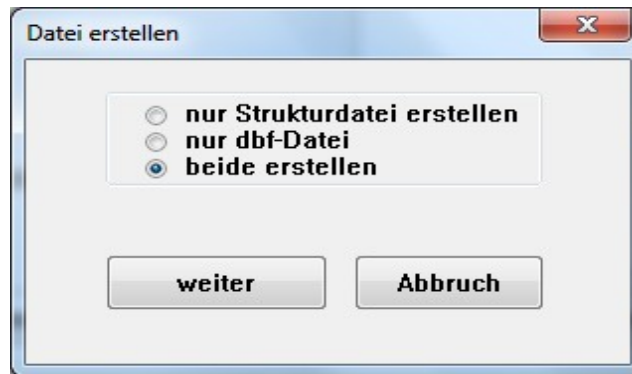


Bild6.

Hier kann entschieden werden ob aus dem eingegebenen Satzaufbau eine Strukturdatei und/oder eine leere DBF Datei erstellt werden soll.

Mit **weiter** erscheinen dann 1 oder 2 Dialoge zur Auswahl unter welchem Namen die DBF/STR Datei(en) zu speichern ist/sind.

[zurück zur Funktionsübersicht](#)

Strukturdatei aus einer bestehenden dbf-Datei erstellen

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der dBase Datei

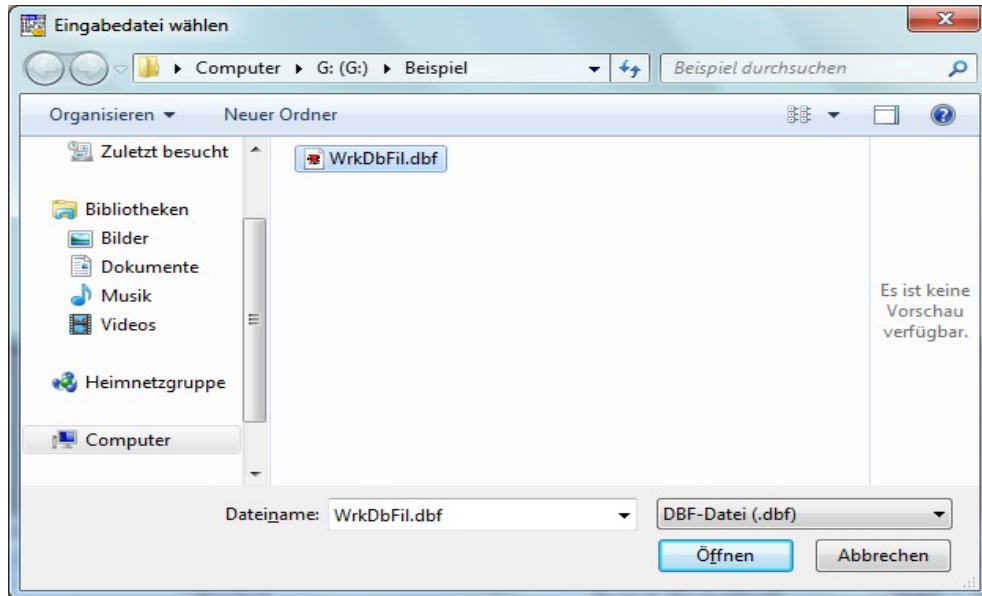


Bild7.

Nachdem die dBase Datei ausgewählt wurde, wird der Dialog zur Auswahl/Eingabe des Namens für die zu erstellende Strukturdatei angezeigt.

Wird eine Strukturdatei ausgewählt, die bereits existiert, erscheint eine Warnung und es besteht die Möglichkeit hier den Vorgang abubrechen.

[zurück zur Funktionsübersicht](#)

Strukturdatei mit Notepad bearbeiten

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der Strukturdatei.

Danach wird NotePad geöffnet und es kann die Strukturdatei bearbeitet werden.

Es ist immer darauf zu achten, dass der Satzaufbau korrekt eingehalten wird.
Zwischen den 4 Feldern muss ein Semikolon (;) stehen.

Das erste Feld enthält den Namen des DBF-Feldes (Großbuchstaben und maximal 10 Stellen)

Das zweite Feld beschreibt die Art des Feldes (**C**, **N**, **D**, **L** oder **M**)

Im dritten Feld wird die Länge des Feldes eingetragen

Im vierten Feld wird die Anzahl der Dezimalstellen eingetragen

Mit der Beendigung von Notepad wird automatisch zum Menü WrkDbFil zurückgekehrt.

Die Benutzung des Notepads zum Korrigieren einer bestehenden Strukturdatei ist dann sinnvoll, wenn die Funktion **Daten einer dbf-Datei in eine andere dbf-Datei kopieren** verwendet wird.

[zurück zur Funktionsübersicht](#)

Indexdatei zu einer bestehenden dbf-Datei erstellen

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der dBase Datei

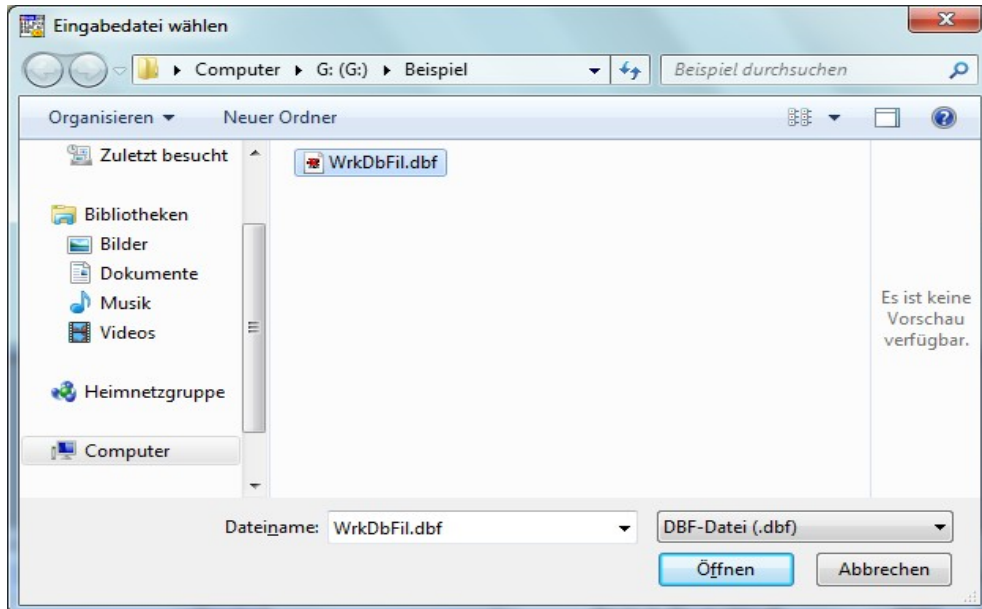


Bild8.

Nachdem die dBase Datei ausgewählt wurde, wird der Dialog zur Auswahl der Schlüsselfelder angezeigt.



Bild 9.

Es sind die Felder die als Schlüsselfelder dienen sollen im linken Fenster zu markieren. Jeder markierte Eintrag wird sofort in das rechte Fenster verschoben:
Die Reihenfolge der Schlüsselfelder ist wichtig!

Die maximale Länge aller Schlüsselfelder konnte nicht geklärt werden, dürfte aber mit maximal 120 Stellen begrenzt sein.

Nachdem die Schlüsselfelder ausgewählt wurden wird der Vorgang mit **weiter** abgeschlossen. In dem nun angezeigten Dialog ist der Name der Indexdatei festzulegen (mit oder ohne Erweiterung .ndx - die Indexdatei erhält immer die Endung .ndx).

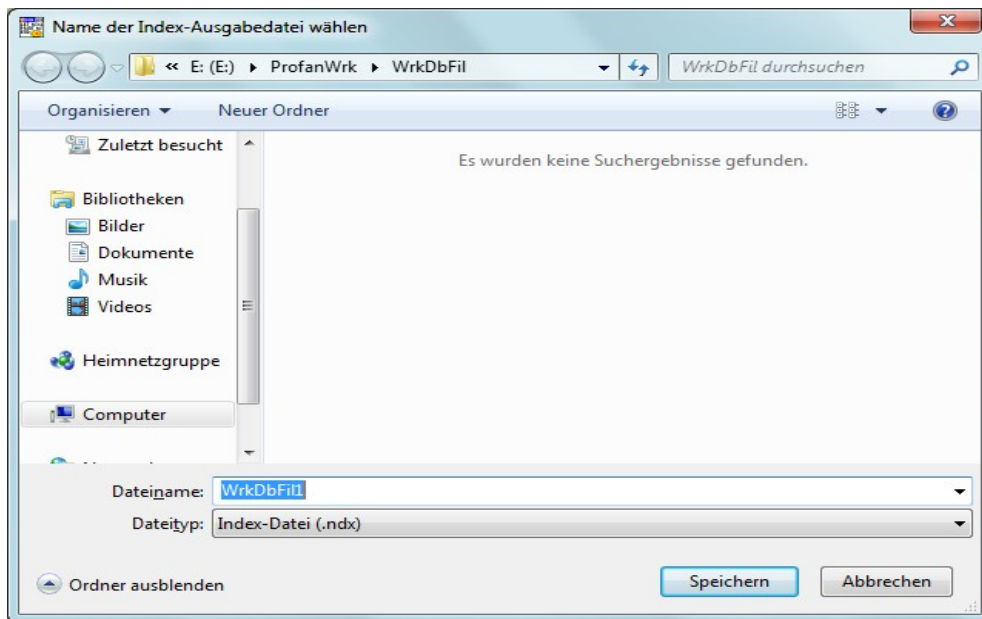


Bild 10.

Wird eine Datei ausgewählt, die bereits existiert, erscheint eine Warnung und es besteht die Möglichkeit hier den Vorgang abubrechen.

[zurück zur Funktionsübersicht](#)

Bestehende dbf-Datei leeren (auf 0 Sätze stellen)

Mit dieser Funktion wird eine dBase Datei geleert, das heißt auf Null Sätze gestellt. Falls auch eine Memodatei existiert wird diese ebenfalls geleert.

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der dBase Datei

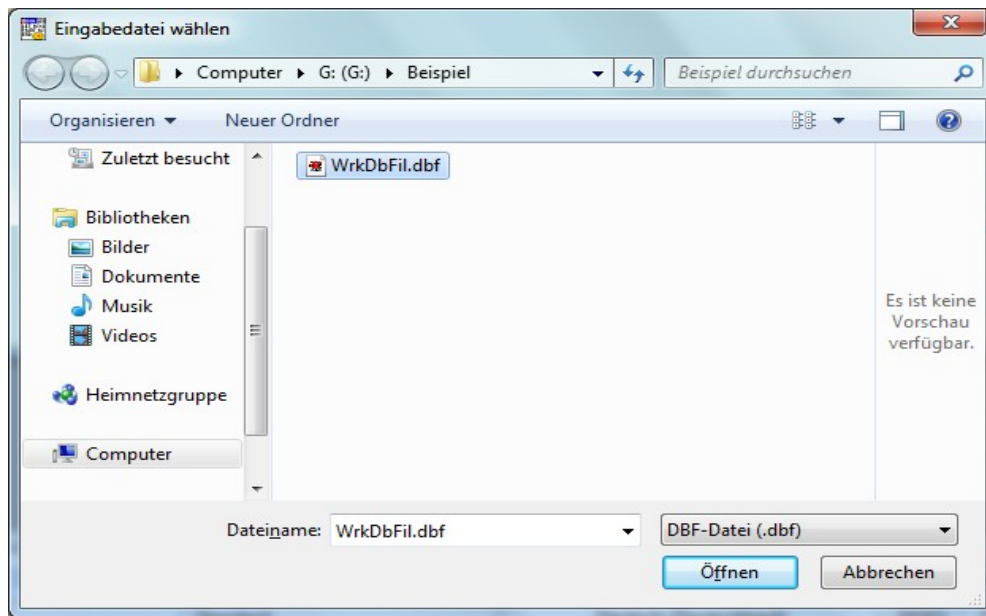


Bild11.

Nachdem die Datei ausgewählt wurde ist mit **Öffnen** zu bestätigen.

Es wird nun die dBase Datei und - wenn gegeben - die Memodatei geleert.

Es gibt hier allerdings ein Problem!

Im dBase III Konzept ist nicht vorgesehen dass in einer dBase Datei auf die Indexdatei referenziert wird – und umgekehrt.

Das heißt: es ist automatisch weder die Möglichkeit gegeben festzustellen ob über einer dBase Datei Indexdateien aufgebaut sind, noch ist erkennbar zu welcher dBase Datei eine Indexdatei gehört.

Daher muss nach dem leeren jede zugehörige Indexdatei separat wieder neu aufgebaut werden.

[zurück zur Funktionsübersicht](#)

Importieren einer CSV-Datei

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der [CSV-Datei](#)

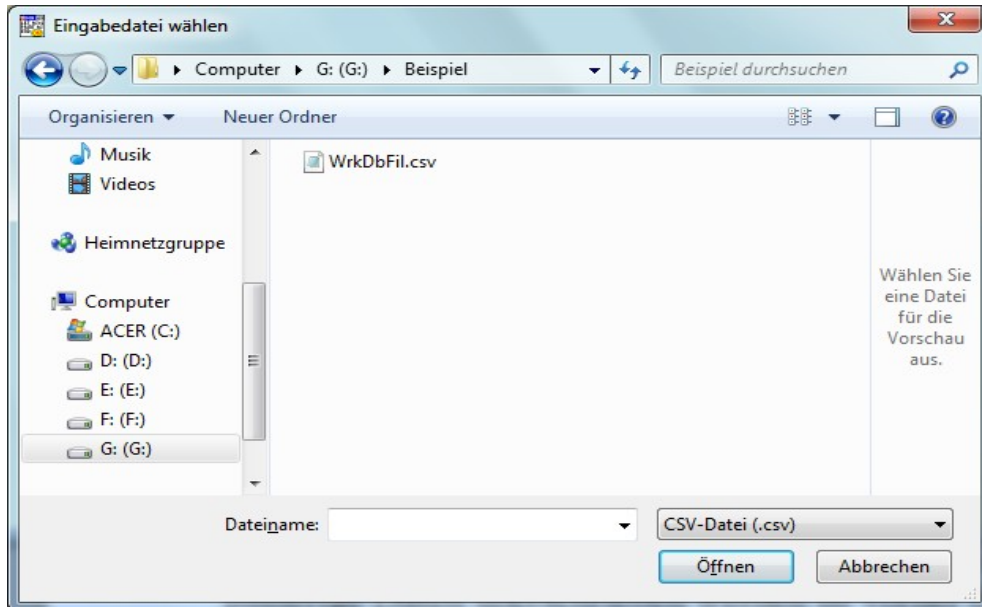


Bild12.

Nachdem die CSV-Datei ausgewählt wurde, wird der Dialog zur Auswahl des Namens für die zu befüllende dBase Datei angezeigt.

Die dBase Datei muss bereits bestehen! Die neuen Daten werden immer hinzugefügt. Sollen bisherige Daten überschrieben werden ist vorher die Auswahl **dBase Datei leeren** auszuführen.

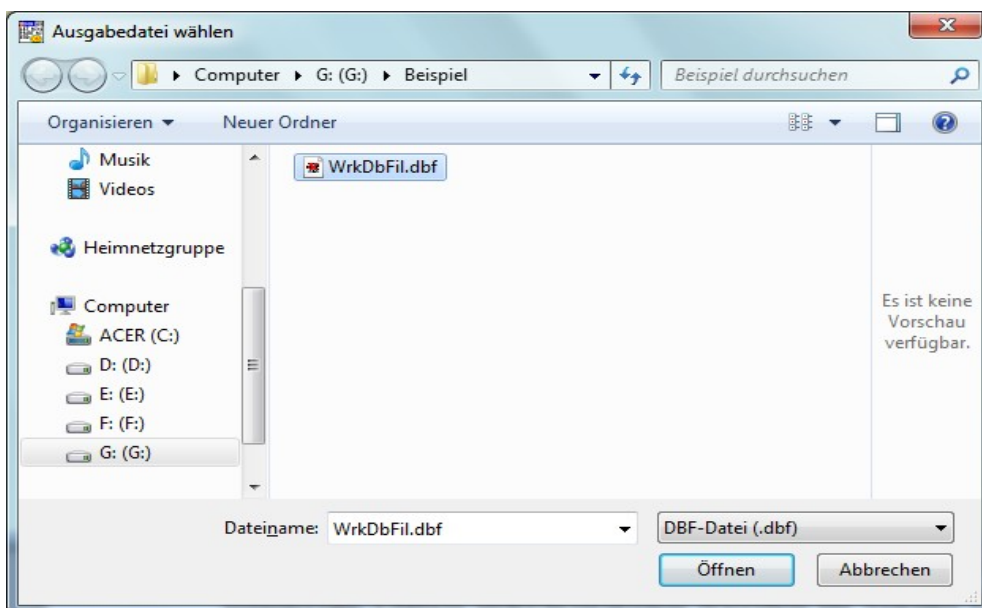


Bild13.

Es werden alle Felder aus der CSV-Datei übernommen. Das heißt es können nur höchstens so viele Felder übernommen werden als Felder im dBase-Satz enthalten sind. Hat der CSV-Satz weniger Felder als der dBase-Satz bleiben die restlichen dBase Felder leer.

Numerische Felder erhalten die Werte rechtsbündig, mit führenden Leerstellen. Bei Dezimalstellen wird Komma-gerecht ausgerichtet (als Komma muss ein Dezimalpunkt gesetzt sein wie er für dBase-Dateien vorgesehen ist). Sollte der Wert mehr Stellen haben als im Feld vorgesehen, wird der ganze Wert links und der Dezimalwert rechts abgeschnitten.

Weitere Prüfungen erfolgen nicht. Die Reihenfolge der CSV-Felder muss in der Feldart genau der Reihenfolge im dBase Satz entsprechen.

Handelt es sich um ein Memofeld, wird der Feldinhalt in die Memodatei übertragen.

Im dBase III Konzept ist nicht vorgesehen dass in einer dBase Datei auf darüber aufgebaute Indexdateien referenziert wird.

Daher muss jede Indexdatei separat wieder neu aufgebaut werden (mit der Funktion **Indexdatei zu einer bestehenden dBase Datei erstellen**).

[zurück zur Funktionsübersicht](#)

Exportieren in eine CSV Datei

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der dBase Datei

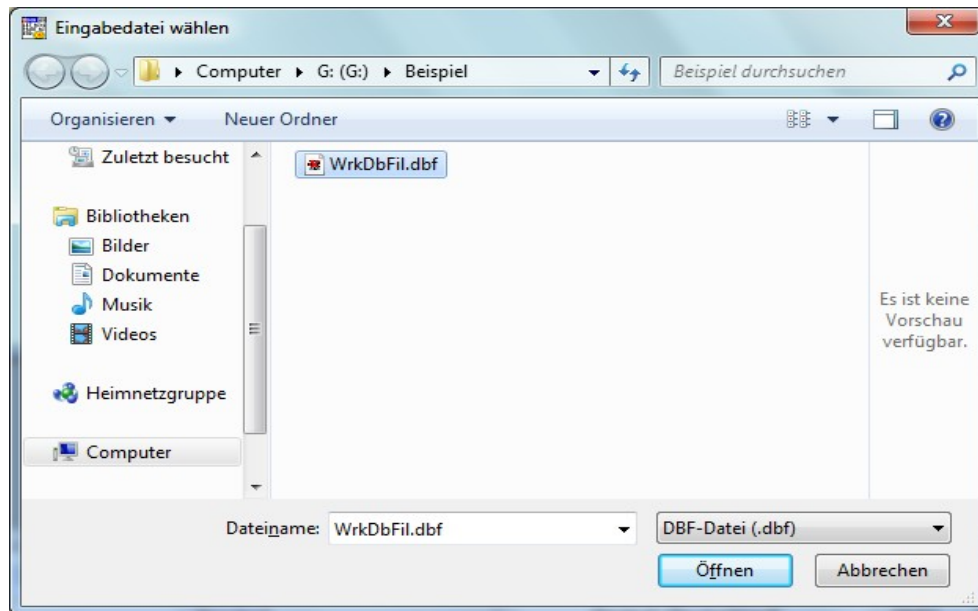


Bild14.

Nachdem die dBase Datei ausgewählt wurde, wird der Dialog zur Auswahl/Eingabe des Namens für die CSV-Datei angezeigt

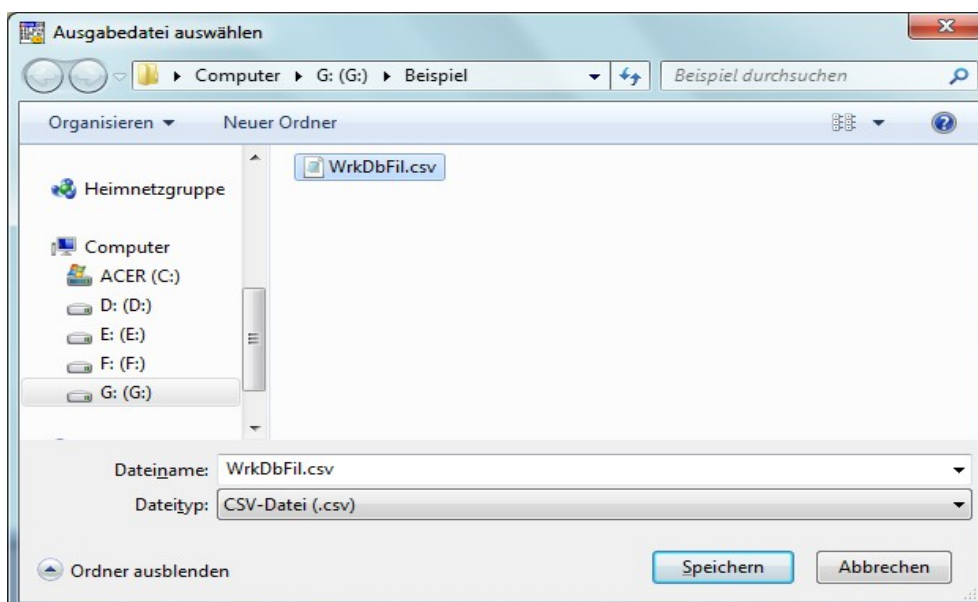


Bild15.

Wird eine Datei ausgewählt, die bereits existiert, erscheint eine Warnung.

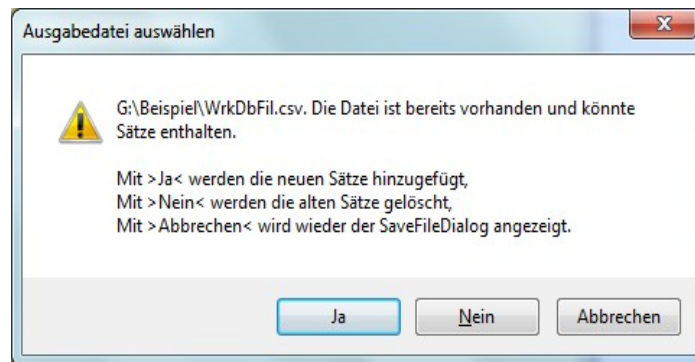


Bild16.

Es besteht die Möglichkeit den Vorgang abubrechen, mit den neuen Daten die bisherigen Daten zu überschreiben, oder die neuen Daten zu den bestehenden Daten anzuhängen.

[zurück zur Funktionsübersicht](#)

Kopieren in eine andere dbf-Datei

Mit dieser Funktion werden die Sätze einer dbf-Datei (der Sendedatei) zu einer anderen dbf-Datei (der Empfangsdatei) hinzugefügt.

Die Empfangsdatei muss bereits existieren.

Sollte die Empfangsdatei nicht existieren so kann sie, für dieses Beispiel mit dem Menüpunkt **dBase Datei aus einer Strukturdatei erstellen** mit der Strukturdatei wrkDbFilN.str , erstellt werden.

Sollen eventuell vorhandene Sätze in der Empfangsdatei ersetzt werden, muss die Empfangsdatei zuvor geleert (auf 0-Sätze gesetzt) werden.

Es werden nur Feldinhalte der Sendedatei in gleichnamige Felder der Empfangsdatei übernommen (Mapping).

Felder die nur in der Empfangsdatei existieren bleiben leer, bei logischen Feldern wird F (false) eingesetzt, numerische Felder werden Komma-gerecht auf Null gesetzt.

Felder die nur in der Sendedatei existieren werden ignoriert!

Die Reihenfolge der (gleichnamigen) Felder kann zwischen den beiden Dateien beliebig sein.

Ist die Feldart bei einem gleichnamigen Sendefeld gleich der Feldart im Empfangsfeld, wird der Inhalt direkt übernommen. Ist das Empfangsfeld kürzer wird bei Zeichenfeldern rechts abgeschnitten. Bei numerischen Feldern kann das Empfangsfeld unterschiedliche Länge und auch eine unterschiedliche Anzahl von Dezimalstellen haben. Der Wert wird immer Komma-gerecht übernommen. Das heißt, es werden Dezimalstellen abgeschnitten (z. B. das Empfangsfeld hat 0 oder weniger Dezimalstellen) oder im Empfangsfeld werden Dezimalstellen ergänzt (Empfangsfeld hat mehr Dezimalstellen als das Sendefeld).

Ist der ganze numerische Wert größer als das Empfangsfeld an ganzen Stellen Platz bietet, wird das Empfangsfeld mit 9 gefüllt. Abgesehen von diesem Fall werden alle anderen Daten kommentarlos beschnitten, wenn das Empfangsfeld kleiner ist.

Nach dem Aufruf dieser Funktion erscheint der Dialog zur Auswahl der 1. dBase Datei (der Sendedatei)

Nachdem die dBase Datei ausgewählt wurde, wird der Dialog zur Auswahl des Namens für die 2. dBase Datei (der Empfangsdatei) angezeigt

Da die Empfangsdatei schon existieren muss werden die neuen Sätze immer zu eventuell schon existierenden Sätzen hinzugefügt. Daher muss die Datei gegebenenfalls zuvor geleert werden.

Es ist auch möglich, dass das Empfangsfeld ein anderer Typ als das Sendefeld ist.

Dabei gelten folgende Regeln:

Sendefeld	Empfangsfeld	Aktion
C	L	Das Empfangsfeld wird auf F gesetzt
C	M	Der Text wird als Memo übertragen
C	N	Wenn die Zeichenfolge eine gültige Zahl ist wird übertragen, sonst wird auf Null gesetzt
D	C	Wird übernommen
D	L	Das Empfangsfeld wird auf F gesetzt
D	M	wird unterdrückt (kein Memotext wird erstellt)
D	N	Nur wenn Empfangsfeld 0 Komma- und >7 ganze Stellen hat. Sonst wird auf Null gesetzt
L	C	wird unterdrückt
L	D	wird unterdrückt
L	M	wird unterdrückt (kein Memotext wird erstellt)
L	N	Feld wird auf Null gesetzt
M	C	Der Memotext wird, soweit möglich, in das Feld übernommen
M	D	wird unterdrückt
M	L	Das Empfangsfeld wird auf F gesetzt
M	N	Feld wird auf Null gesetzt
N	C	Wert wird linksb. übertragen
N	D	wird übernommen, wenn es sich um ein gültiges Datum handelt
N	L	Das Empfangsfeld wird auf F gesetzt
N	M	wird unterdrückt

Bild17.

Tipp: Es kann durchaus sinnvoll sein die Empfangsdatei unter zu Hilfenahme der schon bestehenden Strukturdatei der Sendedatei zu erstellen. Dies kann mit dem Menüpunkt Strukturdatei mit NotePad bearbeiten erfolgen. Das heißt mit dem Notepad die Sende-Strukturdatei auswählen, diesen Satzaufbau an die Bedürfnisse der Empfangsdatei anpassen (Feldlängen ändern, Felder weglassen, neue Felder einfügen), dann als neue Strukturdatei abspeichern und mit dem Menüpunkt **dBase Datei aus Strukturdatei erstellen** die Empfangsdatei erstellen.

[zurück zur Funktionsübersicht](#)

Was ist eine CSV-Datei?

Eine CSV-Datei ist eine gewöhnliche Textdatei die pro Satz die Inhalte aller Felder eines Datensatzes enthält.

Für diese Dateiform haben die meisten Datenbankprogramme eine Import- und Export Funktion.

Es gibt jedoch verschiedenste Abarten dieser Textdatei-Form.

In dieser Datei gibt es weder eine feste Struktur, noch sind die einzelnen Felder beschrieben. Die Inhalte der einzelnen Felder sind im CSV-Satz als Substrings aneinandergereiht. Als Trennzeichen kann ein Beistrich (Komma) oder ein Strichpunkt (Semikolon) - auch gemischt - verwendet werden.

Die Reihenfolge der Substrings muß mit der Reihenfolge der Felder in der Datenbankdatei in die importiert wird übereinstimmen!

Die einzelnen Feldinhalte können innerhalb von Anführungszeichen (") (Gänsefüßchen, double quotes) eingeschlossen sein.

Enthält ein Feld ein Komma, Semikolon oder ein Anführungszeichen muß der Feldinhalt innerhalb Anführungszeichen eingeschlossen stehen!

Sollte innerhalb eines Textes ein Anführungszeichen enthalten sein, ist dieses Zeichen doppelt anzugeben

Beispiele:

A. wie der Text in der Datei aussehen soll

B- wie der Text in der CSV-Datei als Substring stehen muss

Ein Text ohne Beistrich, Strichpunkt und ohne Anführungszeichen

A. **Ein ganz normaler Text:**

B- **Ein ganz normaler Text:**

Beistrich und Strichpunkt einschließen, sonst würde folgender Text als 3 Substrings interpretiert

A: **Ein, Text; mit Beistrich und/oder Strichpunkt**

B- **"Ein, Text; mit Beistrich und/oder Strichpunkt"**

Text mit Anführungszeichen. Jedes Anführungszeichen im Text muss doppelt angegeben sein:

A. **Das Bild "Maxi.jpg" ist bunt**

B- **"Das Bild ""Maxi.jpg"" ist bunt"**

Alle anderen Zeichen sind unproblematisch. Solche Texte können, müssen aber nicht, innerhalb Anführungszeichen eingeschlossen sein:

A. **Abcde fgh**

B- **Abcde fgh ijKLM**

oder

B- **"Abcde fgh ijKLM"**

Punkte (Dezimalpunkte), Bindestriche etc. werden wie normale Zeichen behandelt:

A. **-125.13**

B- **-125.13**

-oder

B. **"-125.13"**

Anmerkungen:

Da CSV-Dateien auch zum Datenaustausch von "nicht-Windowssystemen" benutzt werden können (z.B. Linux) kann es passieren, dass die Textdatei nicht der Windowsversion entspricht- Das heißt: jeder Satz ist mit einem **CRLF** abgeschlossen- **CRLF** (Carriage return + Line feed) bedeutet in der Profanschreibweise **@Char\$(13)+@Char\$(10)** oder auch **\$OD0A-**
Es kann dann ein Satz eventuell nur mit **CR** oder nur mit **LF** abgeschlossen sein.

Profan ab 7.x aufwärts verarbeitet diese Formen einwandfrei. Profanversionen vor 7x tun dies nur teilweise.

Sollte also ein Problem beim Einlesen einer CSV-Datei auftauchen könnte eine solche Datei vor der Verwendung mit meiner **LeseText-dll** (ebenfalls auf meiner H.P. zu finden) in die richtige Form umkopiert werden.

Da zum Lesen und Schreiben der CSV-Datei Profanfunktionen benutzt werden, werden die Sätze in eine Stringvariable eingelesen bzw. aus einer Stringvariablen geschrieben. Hier ist auf die Einschränkung zu achten, dass vor Profan 7.x eine Stringvariable maximal 255 Zeichen enthalten kann. Somit kann in diesem Fall die Satzlänge höchstens 255 Stellen sein.

dBase III erlaubt Zeichenfeldlängen von maximal 254 Stellen.

[zurück zu Daten von CSV-Datei importieren](#)
[zurück zur Funktionsübersicht](#)

Was ist eine Strukturdatei?

Der Aufbau einer Strukturdatei.

Eine Strukturdatei ist eine gewöhnliche Textdatei die pro Satz ein Feld einer dbf-Datei beschreibt. Die Reihenfolge der Feldbeschreibungen bestimmt den Satzaufbau der dbf-Datei.

Eine Feldbeschreibung besteht aus den folgenden 4 Elementen die durch Semikolons (;) getrennt werden:

- Feldname: der Name muss mit einem Buchstaben beginnen. die folgenden Stellen dürfen Buchstaben, Ziffern oder einen Unterstrich enthalten- Der Feldname hat eine Länge von 1 bis 10 Zeichen
- Feldart: damit wird die Art des zu erstellenden Feldes festgelegt
 - C Zeichenfeld: in das Feld dürfen alle Zeichen eingegeben werden
 - N numerisch: in dem Feld sind nur Ziffern, ein Dezimalpunkt oder ein Minuszeichen erlaubt
 - D Datumsfeld
 - L Logik Feld
 - M Memo Feld
- Feldlänge: Die Länge des Feldes ist bei einem Zeichenfeld 1 bis 254, bei einem numerischen Feld 1 bis 19 Stellen. Ein Datumsfeld hat immer eine Länge von 8 Stellen, ein Logikfeld die fixe Länge von 1 Stelle und ein Memofeld ist immer 10 Stellen lang.
- Anzahl Dezimalstellen: nur für numerische Felder. Der Wert kann 0 bis 15 sein. Maximal kann aber die Anzahl von Dezimalstellen nur Feldlänge minus 2 sein! Bei allen nicht-numerischen Feldarten wird hier 0 angegeben.

[Zurück zu dBase Datei aus einer Strukturdatei erstellen](#)
[zurück zur Funktionsübersicht](#)

Anhang A.

Im Programm wrkDbFil wird auch die FileDialog.dll verwendet.

Diese DLL kann sich im selben Verzeichnis von wrkDbFil befinden.

Da diese DLL aber möglicherweise auch in anderen Programmen benutzt wird erscheint es zweckmäßig diese DLL in einem eigenen Verzeichnis (neben anderen DLLs) zu speichern.

Da der Speicherort (das Verzeichnis mit der DLL) im Programm wrkDbFil nicht vorhergesehen und daher nicht definiert werden kann gibt es ein Hilfsmittel: die Datei Profan_Pfade.INI.

Diese Datei ist im Rootverzeichnis auf dem Laufwerk in dem sich auch das Windowsverzeichnis befindet (normalerweise ist das C:\) zu speichern.

Damit kann gesteuert werden wo wrkDbFil diese FileDialog.dll suchen muss. Zuerst wird im Verzeichnis von wrkDbFil gesucht. Wird die DLL dort nicht gefunden wird der Verweis in der Datei Profan_Pfade.INI gesucht.

Im Paket liegt eine Musterdatei Profan_Pfade.INI. Diese ist in das Verzeichnis C:\ zu kopieren (genauer auf das Laufwerk auf dem sich auch das Windowsverzeichnis befindet).

Im Paket liegt eine Musterdatei Profan_Pfade.INI. Diese Datei ist nach der Installation von wrkDbFil nach C:\ zu verschieben (genauer auf das Laufwerk auf dem sich auch das Windowsverzeichnis befindet) bzw. falls dort bereits eine Datei Profan_Pfade.INI existiert entsprechend richtig zu stellen bzw. zu ergänzen.

Mit dem ebenfalls beigefügten Programm updat_Pfade.exe kann in der Datei Profan_Pfade.INI der Pfad zur FileDialog.dll richtig gestellt werden. Alle anderen Einträge in der Datei haben keine Bedeutung – sie könnten auch gelöscht werden (muss aber nicht sein).

Sollte sich die FileDialog.dll im Verzeichnis von wrkDbFil befinden, ist die Datei Profan_Pfade.INI nicht erforderlich.

[zurück zur Funktionsübersicht](#)