SWT
SOFTWARE TOOLS

Deutsches Anwenderhandbuch

Copyright © (1990-2022) SW-Tools ApS
Duevej 23
DK-2680 Solrød Strand
Denmark
Phone: **+45) 33 33 05 56**
Mail: **swtools@swtools.com**
www: **www.swtools.com**

Data-Dictionary

22/11/01 / 2022-09-01 008.384

Inhalt

Inhalt 2
1. Generelle Hinweise 8
1.1. Lizenzinformationen 9
1.2. Start des Data-Dictionary Programms 10
1.3. Anwender Schnittstelle 11
1.3.1. Funktionen für die Tabellendefinition und -erweiterung 13
2. Installation der Treiber 14
2.1. Installation der Treiber 15
3. Treiber Schnittstelle 16
3.1. Schnittstellen Parameter 17
3.1.1. Typ 18
3.1.2. Interner Typ 19
3.1.3. Beschreibung 20
3.1.4. Datenbank 21
3.1.5. Anwendername und Kennwort für ODBC Datenbanken 22
3.1.6. Datenquelle, Zugriffsart und Server-Kennwort 23
3.1.7. Zeichenkonvertierung - sprachenabhängige und spezielle Zeichen 24
3.1.8. Zugriff nur lesend 25
3.2. Besondere Treiber Optionen 26
3.3. Löschen eines installierten Treibers 27
4. Import von ODBC Definitionen 28
4.1. Import von ODBC Tabellendefinitionen in das Data-Dictionary 29
4.1.1. Treibertyp 30
4.1.2. Id 31
4.1.3. Tabellenname 32
4.1.4. Tabellen-Id 33
4.1.5. Generierte ID 34
5. Standard Definitionen 35
5.1. Laden der Standard Tabellendefinitionen aus einer Textdatei 36
5.1.1. Treiber Typ 37
5.1.2. Id 38
5.1.3. Tabellenname 39
5.1.4. Dateiname 40
5.1.5. Generierte ID 41
5.2. Laden der Standard Dateiverknüpfungen 42
5.3. PUT von Tabellendefinition in eine Textdatei 43
5.4. PUT von Dateiverknüpfungen in eine Textdatei. 44
6. Tabellen Informationen 45
6.1. Anlage einer neuen Tabelle 46
6.2. Tabellen Parameter 47
6.2.1. Id 48
6.2.2. Typ 49
6.2.3. Name 50
6.2.4. Wirklicher Name 51
6.2.5. Text 52
6.2.6. Dokumentation 53
6.2.7. Standard ID 54
6.3. Anfrage auf Databankverknüpfungen 55
6.4. Löschen von Tabellen 56
6.5. Überblick über Datensätze 57
6.6. Reservierte Worte 58
7. Feldbeschreibungen 59

7.1. Feldnummer 60
7.2. Feldname 61
7.3. Format 62
7.3.1. Ausgabe numerischer Felder 63
7.3.2. Fließendes Dezimalkomma 64
7.3.3. (nnn) Tabellenfelder 65
7.3.3.1. (nnn+) Feldgruppen in Tabellen 66
7.3.4. W Arbeitsfelder 67
7.3.5. (nn) Explizite Satzlänge 68
7.3.5.1. (nn,x) Explizite Satzlänge in Bits 69
7.3.6. Bnn Explizite Bytenummer 70
7.3.6.1. Bnnn,x Explizite Bitnummer 71
7.3.7. Tn Fester Feldtyp 72
7.3.8. Unn Spezifikke Kodetabelle für ein einzelnes Feld 73
7.3.9. R Rechtsbündige Anzeige 74
7.3.10. S Stopzeichen (delimiter) im alphanumerischen Feld 75
7.3.11. Fnn Systemfeldadresse 76
7.3.12. K,D - DATAMASTER Indexfelder 77
7.3.12.1. E,I,V,X,Z - DATAMASTER Spezial Indexfelder 78
7.3.13. Cnn - DATAMASTER Feldprüfung 79
7.3.14. Qnnn Explizit SQL Typ 80
7.3.15. Zugriffs Modus 81
7.3.16. Pnn Gepackte Felder 82
7.3.16.1. Pnnnn Explizite Angabe des Packtypes 83
7.3.17. Vxxx Sekundäre Packungstypen 85
7.3.17.1. Parameter für sekundären Packungstyp 86
7.3.17.2. Standard für sekundäre Packung für BASIC Datum 87
7.4. Informationen in der Feldzeile, VIEW-Menü 88
7.4.1. Bytenummer 89
7.4.2. Feldlayout 90
7.4.3. SQL-Namen 91
7.4.4. SQL-Typen 92
7.4.5. C-byte 93
7.4.6. C-Variable 94
7.4.7. BASIC-Variable 95
7.4.8. Datensatz 96
7.4.9. Nächster Satz 97
7.4.10. Satzüberblick 98
7.4.11. Hexadezimaler Dump eines Satzes 99
7.4.12. Dezimaler Dump eines Satzes 100
7.4.13. Ascii Dump eines Satzes 101
7.4.14. Übersicht über Hilfstexte 102
7.4.15. Übersicht über Prüfvorschriften 103
7.4.16. Übersicht über Sondertexte 104
7.4.17. Reservierte Worte 105
7.5. Erweiterung von DATAMASTER Dateien mit dem FDF Modul 106
7.5.1. DATAMASTER Datei Konvertierung von BASIC in ODBC 107
8. Index, Hilfstexte und Prüfvorschriften 108
8.1. Index-Definitionen und Tabellenverknüpfungen 109
8.1.1. Indexname 110
8.1.2. Datei ID 111
8.1.3. Directory 112
8.1.4. Schlüsseldefinitionen 113
8.2. Index Betrachtungen 114
8.2.1. BASIC COMET AF-05000 Beispiel 115

8.2.2. Konstanten und Selektionen 116
8.2.2.1. Mehrfach Konstanten in Indexdefinitionen 117
8.2.3. Packen....118
8.2.4. Satznummern....119
8.2.4.1. Index 0....120
8.2.5. Duplikate zugelassen....121
8.2.6. Nullen Unterdrückung....122
8.2.7. Auffüllen der überzähligen Positionen....123
8.2.8. Teilfelder....124
8.2.9. Schlüssel nicht Bestandteil des Satzes (KNIR)....125
8.2.10. Index Schlüsseldefinitionen, LOGICAL Index....126
8.2.11. Directory 00 Satznummer lesen....127
8.2.12. ACCESS bei Verwendung von SWODBC in nichtindizierten Dateien....128
8.2.13. Unterdrücken von IQ Suche bei speziellem Index (*xx)....129
8.2.14. Unterdrückung des Index-Update für einen speziellen Index....130
8.2.15. Anlegen eines Pseudoindexes (+xx)....131
8.2.16. Index Beispiel....132
8.2.17. Schlüsseloptionen, absteigender Index....133
8.2.18. ODBC Zugriff, Beispiel....134
8.2.19. Index SQL-Name....135
8.2.20. Natürliche Indizes und ORDER BY....136
8.2.21. Absteigende Indizes....137
8.2.22. Spezielle Kodetabelle....138
8.3. Betrachten über Tabellenverknüpfungen....139
8.3.1. Abhängigkeit von Schlüssellänge....140
8.3.2. X* feste Verbindung zu Parameterdateien....141
8.3.3. Dateiverknüpfungs Definitionen....142
8.3.4. Mehrfach Verknüpfungen zur gleichen Datei mit Gross-/Kleinbuchstaben....143
8.3.5. Mehrfach Verknüpfungen zur gleichen Datei durch logischen Index....144
8.3.6. Vorgegebene Verknüpfungen bei READ....145
8.4. Feldbeschreibungen....146
8.4.1. Ausgabe der Feldhilfe....147
8.5. Feldprüfvorschriften....148
8.6. Feld Sondertext....149
8.7. Standard SQL-Namen....150
9. Ausdruck der Dokumentation....151
9.1. Drucker....152
9.2. Optionen....153
10. Grundeinstellungen....155
10.1. Directories....156
10.1.1. Dateidefinitionen....157
10.1.2. Datenbank....158
10.1.3. Spool Directories....159
10.2. Sprache....160
10.3. Vorzeichen....161
10.4. Datum Ausgabeformat....162
10.5. Trennzeichen....163
10.6. Änderungsbestätigungen....164
10.7. Format für interne Datumberechnung....165
10.7.1. Jahr 2000....166
10.7.1.1. Sortieren nach Datum....167
10.7.1.2. DOS Datum....168
10.7.1.3. Jahr 2001 Problem....169
10.7.1.4. Jahr 2005 Problem....170
10.7.1.5. Jahr 2010 Problem....171

10.7.1.6. Jahr 2100 Problem....172
10.7.1.7. Datum Eingabeprüfung....173
10.7.1.7.1. IQ Transaktionswahl, Datumeingabe....174
10.7.1.7.2. CHAIN Funktion, maskierte Datumeingabe....175
10.8. FDF Startparameter....176
11. Subsysteme....177
11.1. Subsystem Menü....178
11.2. Öffnen eines Subsystems....179
11.3. Anlegen eines Subsystems....180
11.3.1. Name....181
11.3.2. Paßwort....182
11.3.3. Firmanummer....183
11.3.4. Adresse der Dateidefinitionen....184
11.3.5. Standardadresse der Datenbank....185
11.3.6. Directory der Listen....186
11.3.7. Datenbankvorgabe....187
11.3.8. Beschreibung....188
11.3.9. Module....189
11.3.10. Typ....190
11.3.10.1. Subsystemtype 'Firma'....191
11.3.10.2. Subsystemtype 'Datadict, Listen und Datasyst'....192
11.3.10.3. Subsystemtype 'Basis'....193
11.3.10.4. Subsystemtype 'Dmsystem'....194
11.3.10.5. Subsystemtype 'Standard'....195
11.4. Dateinamen für Firmen....196
11.4.1. Suchen nach Firmendateinamen....197
11.4.2. DOS Dateinamen für BASIC-Dateien....198
11.5. Kodetabellen....199
11.5.1. Nummern der Kodetabellen....200
11.5.2. Kodetabelle für Sortieren....201
11.6. Systemdateien....202
12. Datenbank Treiber....203
12.1. SSV Treiber für Textdateien....204
12.1.1. Tabellenname....205
12.1.2. Feldformat....206
12.1.3. Indexbeschreibung....207
12.2. ODBC Treiber....208
12.2.1. Kodetabelle....209
12.3. BASIC Treiber, generell....210
12.3.1. Feldformate....211
12.3.2. Gepackte Felder....212
12.3.3. Indexbeschreibungen....213
12.4. BASIC Schnittstellen....214
12.5. X/Basic Treiber....215
12.5.1. Dateiname....216
12.5.2. Firmennummer....217
12.6. CTRAS Treiber für Quattro und Sinix....218
12.6.1. Name und Paßwort des Servers....219
12.6.2. Name und Paßwort des Anwenders....220
12.6.3. Art der Verbindung....221
12.6.4. Quattro/UNIX....222
12.6.5. LU0 Path....223
12.7. X/BASIC Sinix C Interface Library Treiber....224
12.7.1. Art der Verbindung....225
12.7.2. Optionen....226

12.8. Quattro Schnittstelle mit 32 Bit FTP 227
12.8.1. Offline Verarbeitung von Quattro Dateien 228
12.8.2. Dateidefinitionen 229
12.8.3. XBasic und andere Schnittstellen mit 32 Bit FTP 230
12.9. XNet Treiber 231
12.9.1. XNet Dateinamen 232
12.10. Netbasic Treiber 233
12.10.1. Anwender 234
12.10.2. Netbasic Dateinamen 235
12.10.3. NETBASIC für CTRAS Funktionen 236
12.11. UNIBASIC / Surfbasic / CX-Basic / Open-Basic Treiber 237
12.11.1. UNIBASIC Dateien 238
12.12. NAVISION Financials 239
12.12.1. Installation von Navision Financials und C/FRONT 240
12.12.2. Installaion und Konfiguration der Navision Financials Schnittstelle 241
12.12.3. Import der Tabellendefinitionen 242
12.13. CONCORDE C5/XAL 243
12.13.1. Export der Concorde Dateidefinitionen 244
12.13.2. Installation und Konfiguration der Schnittstelle zu Concorde 245
12.13.3. Import der Tabellendefinitionen 246
12.14. Btrieve Version 5.10 und 6.15 247
12.14.1. Installation und Konfiguration der Schnittstelle zu Btrieve 248
12.14.1.1. Erweiterte Optionen 249
12.14.1.1.1. Öffnen einer Datei 250
12.14.1.1.2. Suffix Dateiname 251
12.14.1.1.3. Optionen Dateiname 252
12.14.1.1.4. Parameter 253
12.15. ODBC Treiber Einrichtung 254
12.16. ODBC Treiberschnittstelle - Anwendungstyp 255
12.16.1. Type 1 - Unterstützung der Informix Datenbank Sperrung 256
12.16.2. Type 4 - ORDER BY 257
12.16.3. Typ 4096 - Alphanum. Felder dürfen NICHT leer sein 258
12.16.4. Typ 8192 - ODBC Treiber, die nur eine Tabellenverbindung unterstützen 259
12.17. ISAM Datenbankschnittstelle - Anwendungstyp 260
12.17.1. Typ 2 - Unterstützung des SAMSON Systems 261
13. Sonstige Treiber / Anmerkungen zu einigen Anwenderpaketen 262
13.1. GLOBAL 3000 Speedbase und Global 2000 ISAM/DMAM 263
13.2. ALX Datenbank 264
13.3. SSV Datenbankschnittstelle 265
13.3.1. Schreiben Sortierungsindex 266
13.3.2. Länge des Vorspanns (Header) 267
13.3.3. Erster Datensatz 268
13.3.4. Anlage Indexdatei > Sätze 269
13.3.5. CRLF beim Schreiben 270
13.3.6. Name der Indexdatei 271
13.3.7. Duplikate für >= Index 272
13.3.8. CRLF beim Lesen 273
13.3.9. Dateikomprimierung 274
13.3.10. Start Codetabelle 275
13.4. Erweiterte SSV Schnittstelle 276
13.4.1. Feste Satzlänge 277
13.4.2. Feste Blocklänge 278
13.4.2.1. Maximale Anzahl von Datensätzen 279
13.4.3. Feste Sätze/Block 280
13.4.4. Erstes freier Datensatz 281

13.4.5. Anzahl freier Datensätze 282
13.4.6. Nächster freier Datensatz 283
13.4.7. Dateiname Typ....284
13.4.8. Feldpackung Typen....285
13.4.9. Schlüsselfeldpackung Typen 286
Figuren 287
Index....289

1. Generelle Hinweise

Das Data-Dictionary enthält Informationen und Daten, die für den Aufbau einer Datenbank notwendig sind. Eine Datenbank ist eine Sammlung von Tabellen die gezielt auffindbare Informationen enthält.
Jeder Tabelle ist ein Tabellentyp zugeordnet. Hierbei kann es sich z.B. um den Tabellentyp ODBC, C-ISAM oder BTrieve handeln.
Alle Tabellen beinhalten eine Beschreibung der Tabellenelemente (Felder), jeweils beschrieben durch Nummer, Name und Format. Im Gegensatz zu anderen Datenbank- Systemen sind Leerstellen und Sonderzeichen in den Feldnamen sowie Definitionen logischer Feldformate, die in SW-TOOLS TRIO verwendet werden, zugelassen.
Eine Tabelle beinhaltet auch Index-Definitionen, mit denen die Reihenfolge der Daten in der physikalischen Datei beschrieben wird, z.B. enthält eine Artikeldatei normalerweise als Hauptindex die Artikelnummer, und als Unterindex könnte man sich die Lieferantennummer bzw. -namen vorstellen. Über den Index ist ein direkter und schneller Zugriff auf die gewünschten Daten möglich.
Eine wesentliche Eigenschaft des Data-Dictionary ist die Möglichkeit, Verbindungen zwischen den einzelnen Tabellen festzulegen. Sind diese Verbindungen definiert, stehen in SW-Tools-TRIO die Möglichkeiten wie z.B. graphische Darstellung der Tabellenverbindungen, Verknüpfungen zu Feldern anderer Tabellen, gleichzeitiges Ablaufen interaktiver Anfrageprogramme zur Verfügung.
Das Data-Dictionary verfügt über eine komplette Beschreibung des Datenbanksystems mit den folgenden Eigenschaften:

- **Definition der Tabellen**
- **Definition der Elemente und deren Formate**
- **Definition der Indizes**
- **Definition Verbindungen zwischen den Tabellen**
- **Dokumentation der Tabellen und deren Elemente**

1.1. Lizenzinformationen

Die Data-Dictionary ist ein Copyright geschütztes Produkt von SW-Tools. Die Lizenzinformationen werden bei Programmstart angezeigt.

1. Lizenzinformationen

Die Benutzung von die Data-Dictionary ist nur im Umfang der entsprechenden Lizenzvereinbarung gestattet.

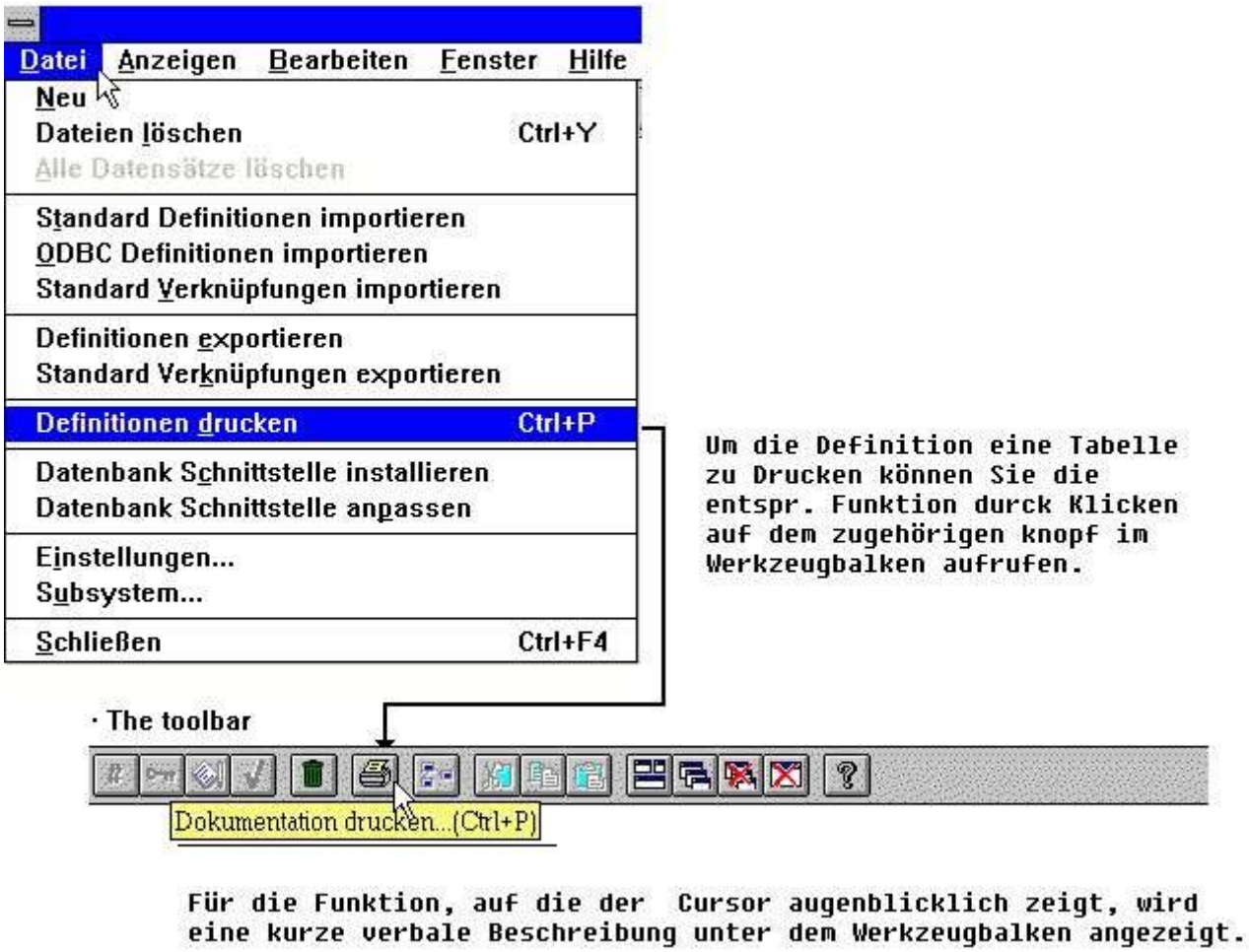
1.2. Start des Data-Dictionary Programms

Nach Start des Data-Dictionary wird folgendes Bild angezeigt:

2. Data-Dictionary

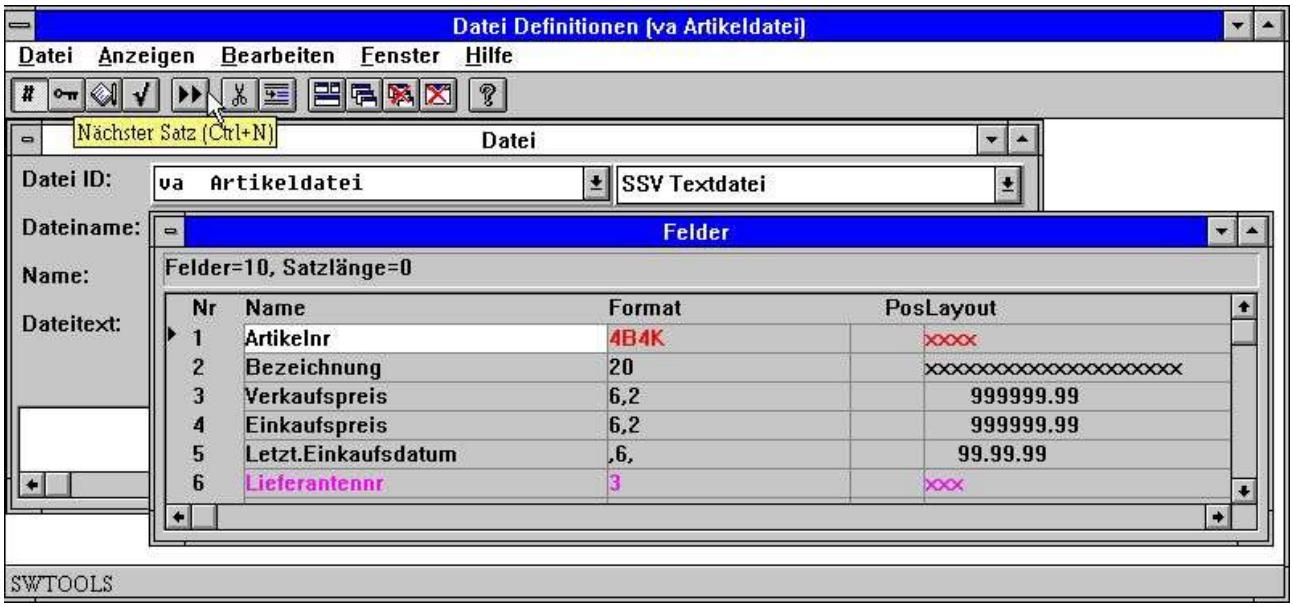
1.3. Anwender Schnittstelle

Um die einzelnen Funktionen im Data-Dictionary zu aktivieren, können Menüs oder Knöpfe in der entsprechenden Symbolleiste benutzt werden.



3. Zugriff zu den Funktionen über Menüs oder Symbolleiste

Da die Symbolleiste nicht alle Funktionen gleichzeitig enthalten kann, ändert sich diese entsprechend den einzelnen Funktionesbereichen (wie die Menüs). Z.B. enthält die Symbolleiste folgende Funktionen bei Aktivierung der Feldfunktionen:



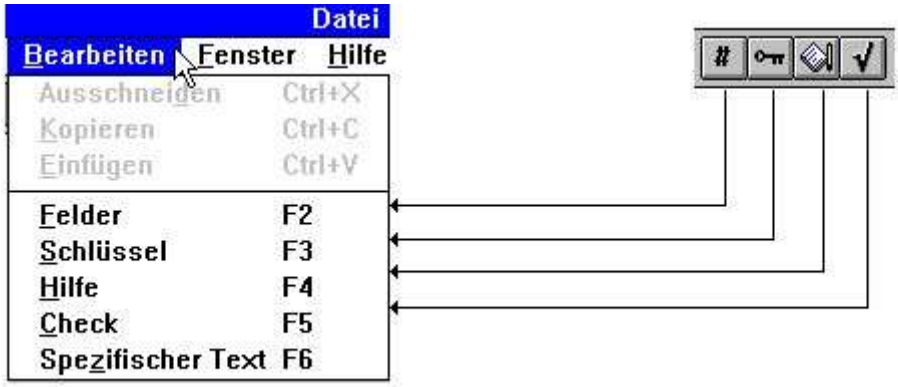
4. Andere Knöpfe in der Symbolleiste

Auch wenn die Symbolleiste sich entspr. den Funktionsbereichen ändert, verbleiben einige generelle Funktionen unverändert. Hierbei handelt es sich die Plazierung des Fensters, Schließen des aktuellen Fensters und Aufruf der Online-Hilfe.
Diese generellen Funktionen sind natürlich auch immer über die Menüs aufrufbar.

5. Generelle Menüs und Funktionen in der Symbolleiste

1.3.1. Funktionen für die Tabellendefinition und -erweiterung

Das Data-Dictionary enthält folgende Funktionen für Tabellenverarbeitung:



6. Funktionen

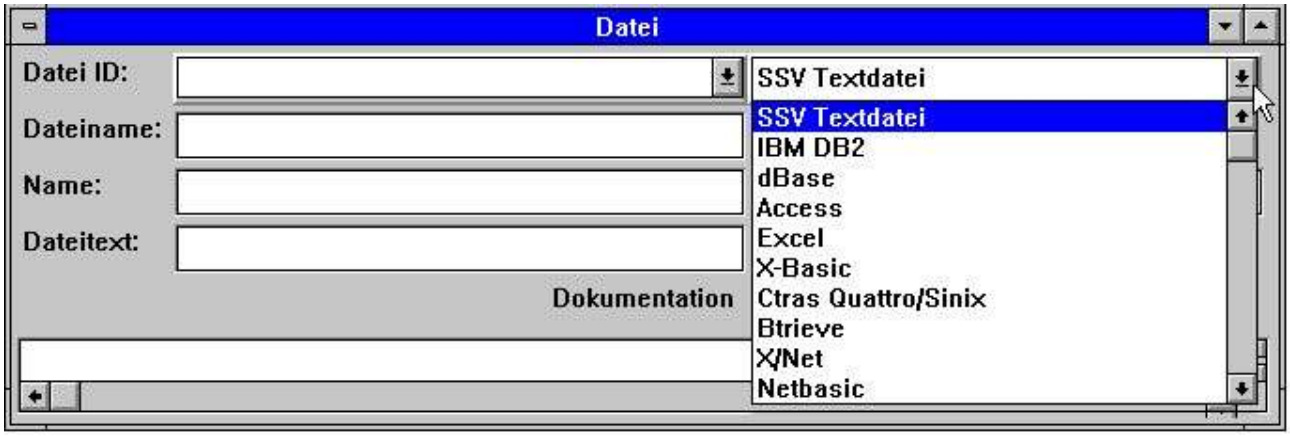
2. Installation der Treiber

Die Funktion für die Installation der Treiber wird über folgendes Menü aufgerufen:

7. Wahl der Funktion für die Installation der Treiber

2.1. Installation der Treiber

Jede im Data-Dictionary definierte Tabelle muß mit einem Treiber verbunden werden. Z.B. sind die Demo-Dateien mit dem internen SW-Tools SSV Texttreiber verbunden (einfache Semikolon getrennte Textdateien).
Die folgende Liste zeigt an, welche Treiber installiert sind:



8. Installierte Treiber

Weitere Treiber können durch Wahl der folgenden Funktion installiert werden:

9. Installation weiterer Treiber

Diese Funktion erlaubt die Installation weiterer Treiber. Ist der gewählte Treiber bereits installiert, wird er durch den neuen erstattet, unter Voraussatzung, daß die Namen jeweils identisch sind. Ist die nicht der Fall, können zwei oder mehrere gleiche Treiber unter verschiedenen zugehörigen Optionen installiert werden.

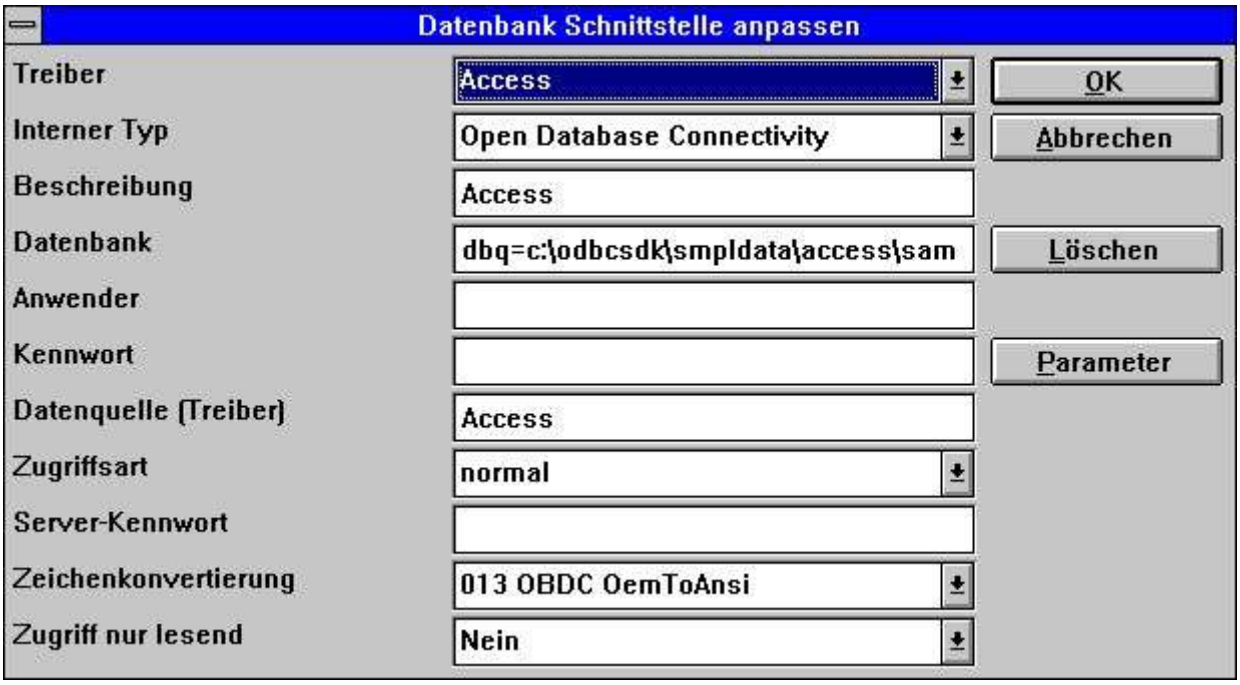
3. Treiber Schnittstelle

Die Treiber Schnittstellenfunktion wird aus folgendem Menü gewählt.

10. Wahl der Treiber Schnittstellenfunktion

3.1. Schnittstellen Parameter

Installierte Treiber werden über eine Reihe von Parameter gesteuert. Identisch für alle Treiber sind Parameter wie z.B. Standard Adresse, Name des Servers, Art der Zeichenkonvertierung (z.B. von UNIX zu DOS). Darüber hinaus können einem Treiber individuelle Parameter zugeordnet werden. Dies geschieht durch Aktivierung besonderer Optionen. Um Standard-Treiberparameter zu ändern, ist folgende Funktion zu wählen.



11. Ändern der Treiberparameter

3.1.1. Typ

Der Dateityp bestimmt den aktuellen installierten Treiber. Bei Wahl eines Treibers wird die Änderung der Parameter für den gewählten Treiber ermöglicht.

3.1.2. Interner Typ

Der interne Dateityp bestimmt den Typ des physikalischen Treibers. Der Wert wird intern verwendet und kann jedem der installierten Treiber entsprechen.

3.1.3. Beschreibung

Die Beschreibung wird bei Auflistung der zur Verfügung stehenden Treiber verwendet. Man sollte deshalb hier einen aussagefähigen Namen wählen.

3.1.4. Datenbank

Eine Datenbank ist eine Sammlung einer oder mehrerer Tabellen, die physikalisch an beliebigen Stellen im System abgelegt sind. Für ODBC kann Optionen als DBQ=c:\Access,OPT=Y angegeben werden.

3.1.5. Anwendername und Kennwort für ODBC Datenbanken

Anwendername und Kennwort können zusammen mit der ODBC Schnittstelle verwendet werden. Alle Treiber, die den internen Typ ODBC haben, verlangen u.U. einen Anwendernamen und ein Paßwort, um sich in der Datenbank anmelden zu können (logon). Weitere Informationen hierzu können Sie im Handbuch 'Systemverwaltung' erhalten.

3.1.6. Datenquelle, Zugriffsart und Server-Kennwort

Der Datenquelle (Servername) und das Paßwort stehen für ODBC und Windows Sockets TCP/IP Treiber zur Verfügung. Der Servername verweist auf den Name des UNIX-hosts, wenn die Verbindungsart Windows Socket TCP/IP ist, und auf die Datenherkunft bei einer ODBC Datenbank.
Die Verbindungsart muß auf 'Normal' für ODBC-Treiber gesetzt werden. Alle anderen Treibern können für Windows Socket TCP/IP gesetzt werden, wenn die Datenbank auf einem UNIX-System installiert ist.
Die Windows Sockets TCP/IP Verbindung verlangt nur einen Servername. Dieser Name kann als IP-Adresse oder, wenn in der HOST-Datei so angegeben, als ein Servername eingegeben werden. Z.B. kann eine IP-Adresse

200.0.0.1

sein. Ist die Verbindungsart Windows Sockets TCP/IP, muß der SW-Tools UNIX Server installiert sein. (Siehe weiter unten)

3.1.7. Zeichenkonvertierung - sprachenabhängige und spezielle Zeichen

Alle Treiber können eine Kodeumwandlungstabelle beim Lesen von Daten in einer Datenbank verwenden.
Ist der Treiber mit Windows Sockets TCP/IP verbunden, und handelt es sich um eine Datenbank, die auf einem UNIX-System installiert ist, ist es nicht sicher, daß sprachabhängige Zeichen und Sonderzeichen ohne Umwandlungstabelle korrekt verstanden werden.
Das Data-Dictionary stellt eine Reihe von Kodeumwandlungstabelle zur Verfügung. Im folgenden hierzu einige Beispiele:

0 Keine
1 UNIX/DOS Dänisch
2 UNIX/DOS Deutsch
10 ISO8850
13 ODBC (OemToAnsi)

3.1.8. Zugriff nur lesend

Einige Treiber unterstützen eventuell nicht das Schreiben in eine Datenbank. Wir verweisen auf die Beschreibung der jeweiligen Treiber. Bitte prüfen Sie, daß ein Schreiben in die Datenbank zugelassen ist.

3.2. Besondere Treiber Optionen

Ein Treiber kann besondere Optionen anbieten, die ausschließlich für diese Schnittstelle zu verwenden sind, z.B. Besitzer einer Tabelle bei ODBC-Treibern oder Firmanummer für Basic Systeme.

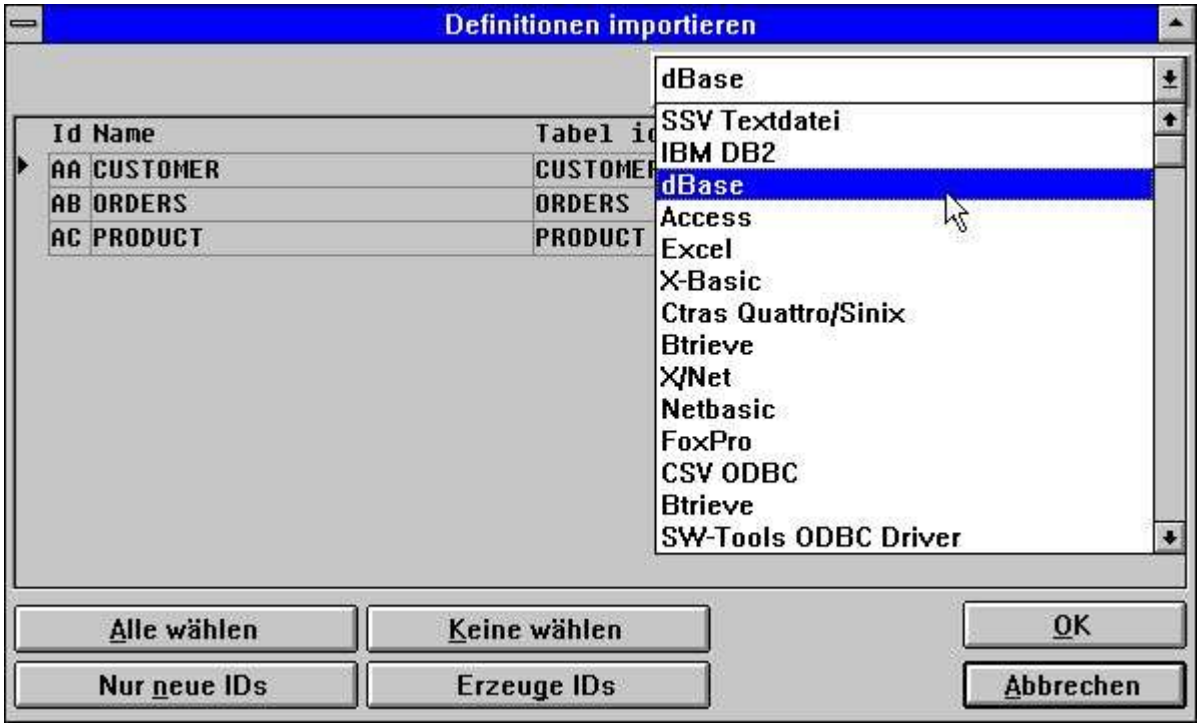
3.3. Löschen eines installierten Treibers

Ein installierter Treiber kann, wenn er nicht mehr gebraucht wird, mit der Löschtaste gelöscht werden.
Diese Funktion löscht nicht die Dateidefinitionen, die mit diesem Treiber verbunden sind. Dies hat manuell zu erfolgen.

4. Import von ODBC Definitionen

4.1. Import von ODBC Tabellendefinitionen in das Data-Dictionary

Nach Installation eines ODBC-Treibers können die Tabellendefinitionen direkt in das Data-Dictionary importiert werden. Z.B. können Tabellen, die in Access oder Excel definiert wurden, für den Gebrauch in SW-Tools-TRIO importiert werden.



12. Import von ODBC Definitionen

4.1.1. Treibertyp

Vor dem Import von Tabellendefinitionen muß ein Treiber gewählt sein. Nach Wahl dieses Treibers werden alle zugänglichen Tabellen, falls welche vorhanden, angezeigt.

4.1.2. Id

Jede Tabelle ist eindeutig durch eine zweistellige ID, die automatisch generiert wird (AA, AB, AC ....), gekennzeichnet.
Sie können auch eine eigene ID angeben. Beachten Sie bitte, daß bestehende Definitionen mit gleicher ID überschrieben werden.
Bei Gebrauch der Funktion 'Nur neue IDs' werden alle vorgeschlagenen IDs, die bereits benutzt sind, übergangen.
Bei Gebrauch der Funktion 'ID Generierung' werden für alle Tabellen neue IDs generiert, wie z.B. A0, A1, A2 ....

4.1.3. Tabellenname

Der Tabellenname ist ein Text und wird vom Treiber übernommen. Vor dem Import kann dieser Name erweitert werden. Der Name wird zusammen mit der ID bei späterem Zugriff auf die Tabelle angezeigt.

4.1.4. Tabellen-Id

Die Tabellen-Id entspricht dem physischen Namen. Für ODBC Tabellen gilt, daß die bei Tabellenanlage angegebene ID den physischen Namen bildet. Die Tabellen-ID kann nur nach Import der Tabellendefinitionen geändert werden.

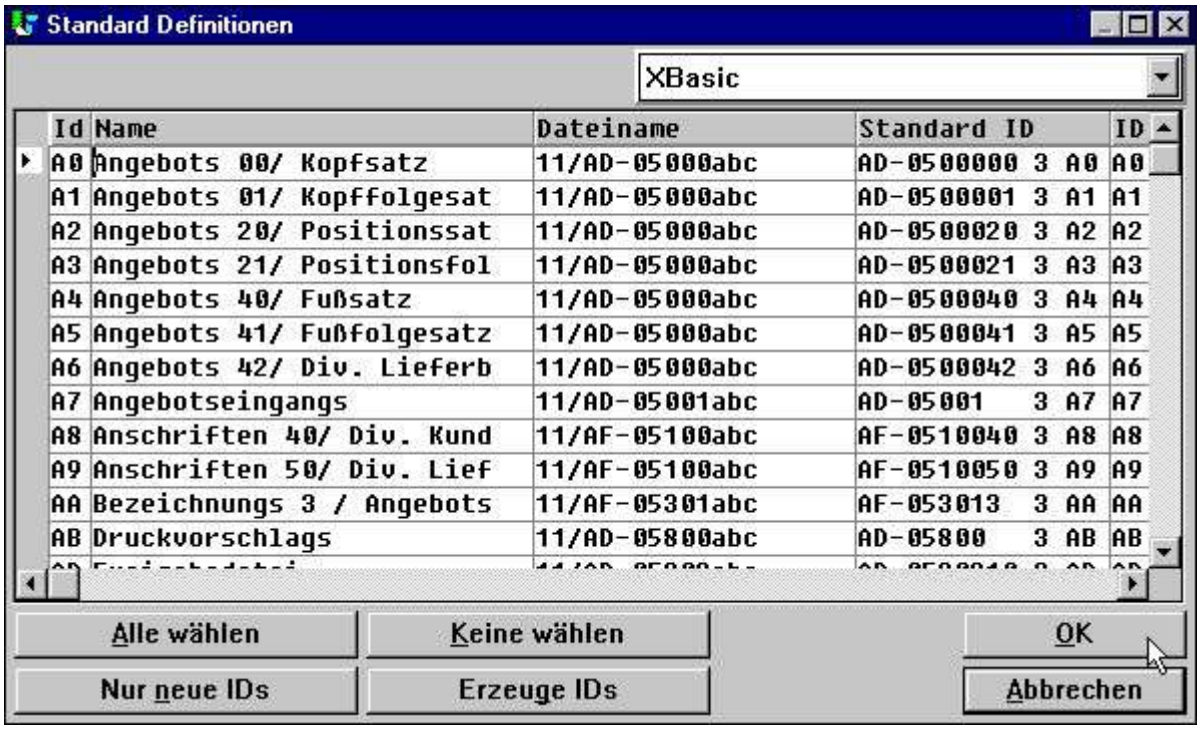
4.1.5. Generierte ID

In dieser Spalte wird die generierte ID angezeigt. Wurde die ID in der ersten Spalte geändert, wird dies hier ignoriert.

5. Standard Definitionen

5.1. Laden der Standard Tabellendefinitionen aus einer Textdatei

Das Data-Dictionary erlaubt das Laden der Standard Tabellendefinitionen aus einer Textdatei. Ist die BASIC Version der SW-Tools Dateidefinitionen bereits installiert, kann in dem Datei Definitionsmodul mit Hilfe des PUT-Befehls eine entsprechende Textdatei generiert werden. Diese Textdatei kann dann mit der folgenden Funktion geladen werden:



13. Standard Definitionen von der BASIC Version der SW-Tools Dateidefinitionen

5.1.1. Treiber Typ

Vor dem Laden der Definitionen muß der Treibertyp angegeben werden. Die Definitionen erhalten dann beim Laden den gleichen Typ.

5.1.2. Id

Die Tabellen aus der Textdatei werden mit der ursprünglichen ID angezeigt, und können vor dem Laden erweitert werden.
Man kann eine neue ID angeben bzw. die bestehende ID durch Leerstellen ersetzen. Die entsprechende Tabelle wird dann übersprungen. Beachten Sie bitte, daß Definitionen mit gleicher ID überschrieben werden.
Bei Gebrauch der Funktion 'Nur neue IDs' werden alle vorgeschlagenen IDs, die bereits benutzt sind, übergangen.
Bei Gebrauch der Funktion 'ID Generierung' werden für alle Tabellen neue IDs generiert, wie z.B. A0, A1, A2 ....

5.1.3. Tabellenname

Der Tabellenname ist ein Text und wird vom Treiber übernommen. Vor dem Import kann dieser Name erweitert werden. Der Name wird zusammen mit der ID bei späterem Zugriff auf die Tabelle angezeigt.

5.1.4. Dateiname

Dies ist der physische Name der Datei, und kann vor dem Laden erweitert werden. Man kann dies z.B. zum ändern der disknumber für eine Gruppe von BASIC-Dateien benutzen.

5.1.5. Generierte ID

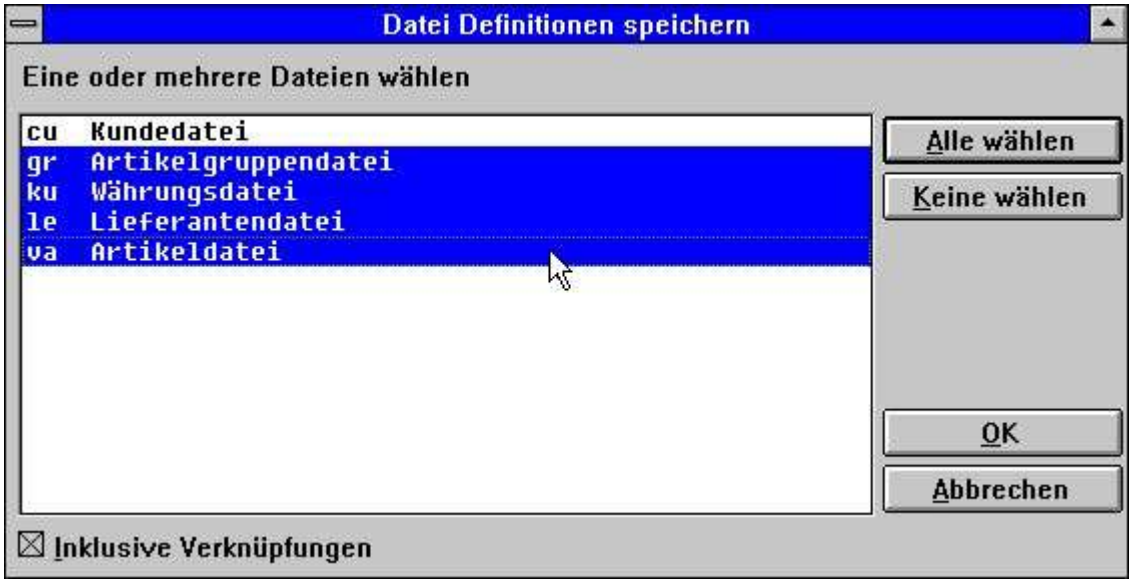
In dieser Spalte wird die generierte ID angezeigt. Wurde die ID in der ersten Spalte geändert, wird dies hier ignoriert.

5.2. Laden der Standard Dateiverknüpfungen

Bei Zurverfügungstellung eines Satzes von Master-Dateidefinitionen, sind die Dateidefinitionen mit der normalen PUT-Funktion abgelegt. Die Dateiverknüpfungen werden jedoch in getrennten Dateien gehalten. Hier behält jede Datei seine ursprüngliche System-ID und eine Standard-ID (SID), mit deren Hilfe die Datei auch nach evt. Änderung der ID identifiziert werden kann.
Beim Laden (LOAD) der Standard Dateiverknüpfungen prüft das System als erstes, welche der Dateien, die nicht aufgrund der Datei ID, sondern aufgrund der ursprünglichen SID aktiv sind. Es werden nur die Verknüpfungen für die aktiven Dateien geladen. Während des Ladens werden die IDs dem aktiven System angepaßt.
Standard Dateiverknüpfungen werden voran im Namen mit einem -=> gekennzeichnet. Beim wiederholten Laden werden alte Verknüpfungen entfernt und durch die aktuellen Verknüpfungen ersetzt. Bitte beachten Sie dies, wenn Sie Dateiverknüpfungen ändern. Wird die Kennung -=> nicht entfernt, werden die durchgeführten Änderungen bei erneutem Laden überschrieben.

5.3. PUT von Tabellendefinition in eine Textdatei

Dateidefinitionen können mit Hilfe der PUT-Funktion in eine Textdatei abgelegt werden. Diese Definitionen können später in einem anderen System über die GET-Funktion aufgerufen werden.



14. PUT Tabellendefinitionen in eine Textdatei

5.4. PUT von Dateiverknüpfungen in eine Textdatei.

Diese Funktion generiert eine Textdatei mit den Verknüpfungen ausgewählter Dateien, und kann später mit der LOAD-Funktion für Standard Dateiverknüpfungen aufgerufen werden.
Beispiel für eine SID: AF-0500002031AS, wobei AF-05000 eine COMET-Datei mit dem Satztyp 020, COMET Version 3.1 und Master Datei-ID AS ist.
In der LOAD und PUT Funktion für Verknüpfungen werden nur die ersten 13 Stellen der SID verwendet. Die Datei-ID wird automatisch gesetzt. Die SID wird, falls keine andere angegeben ist, automatisch entsprechend der Datei-ID gesetzt.

6. Tabellen Informationen

Eine Tabelle enthält die Beschreibung, wie die Daten gespeichert wird. Es wird der Datenbanktyp beschrieben, der physikalische Name und die Adresse der Datei.

6.1. Anlage einer neuen Tabelle

Bei Anlage einer neuen Tabelle muß eine eindeutige ID angegeben werden. Die Regeln hierfür werden später in diesem Abschnitt erläutert.

15. Id für eine neue Tabelle

Soll die neue Tabelle auf einer bereits definierten Tabelle aufbauen, muß die entsprechende Tabelle aus der Liste bestehender Tabellen gewählt werden. Markieren Sie diese Tabelle und wählen Sie die Funktion 'COPY file'. Das Data-Dictionary kopiert jetzt alle Definitionen der Felder, Indizes usw. der markierten Tabelle in die neue Tabelle.

6.2. Tabellen Parameter

6.2.1. Id

Die Tabellen ID muß mit zwei Zeichen angegeben werden, wobei das erste Zeichen ein Buchstabe sein muß, das zweite ein Buchstabe oder eine Ziffer sein kann, z.B. X1.
Die folgenden IDs sind für das System reserviert:

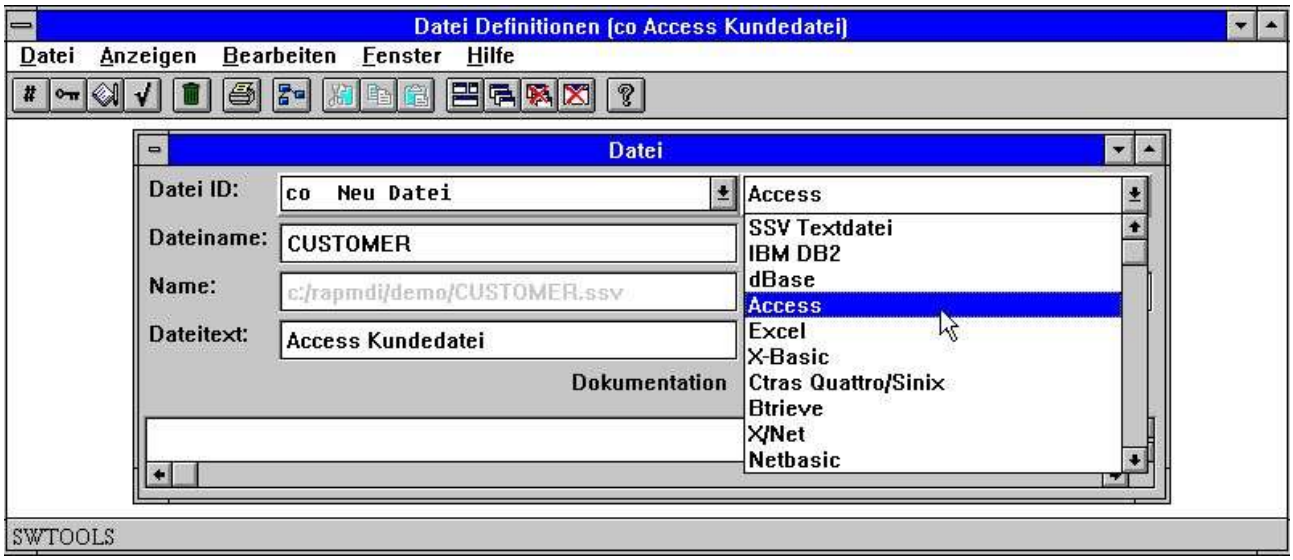
SY Systemfelder
WW Arbeitsfelder

6.2.2. Typ

Die zulässigen Tabellentypen sind von den installierten Treibern abhängig. Das System kann z.B. folgende Typen enthalten:

- **ODBC Treiber (Access,Excel etc.)**
- **C-ISAM**
- **BTrieve**
- **X-Basic**
- **UNIBASIC/SURFBASIC**
- **andre**

Wurde der gewünschte Typ nicht in der Liste gefunden, verweisen wir hier auf den Abschnitt 'Installation der Treiber'.
Soll eine Tabelle den Access ODBC Treiber benutzen, muß folgende Wahl getroffen werden:



16. Wahl des Treibers für eine bestimmte Tabelle

6.2.3. Name

Der Tabellenname ist ein physikalischer Name. Für den Dateityp wie z.B. BTrieve kann dieser als physikalischer Name eingegeben werden wie folgt:

C:/BTRIEVE/CUSTOMER.DAT

oder wenn es eine ODBC Tabelle (oder ähnliche) ist, als nur: CUSTOMER

6.2.4. Wirklicher Name

Basierend auf einem gewählten Tabellentyp und einem Tabellenname wird der wirkliche Name für die Tabelle gefunden. Der angezeigte Name muß der aktuellen Plattendatei der Tabelle auf dem aktuellen Server entsprechen. Andernfalls ist eine Lesen dieser Tabelle nicht möglich.
Der wirkliche Name wird anhand der Informationen für den zugehörigen Treiber und dem Tabellennamen generiert. Handelt es sich um einen X-Basic Treiber, der Standard Adresse

/X.BASIC/0/

und dem Tabellennamen

90/LF-06000abc

wird der wirkliche Name z.B.

/X.BASIC/0/90/LF-06000001

6.2.5. Text

Der Tabellentext ist ein logischer Name für die Tabelle. Der Name kann alle Buchstaben, Ziffern, Leerstellen und Sonderzeichen enthalten. Der Text wird zusammen mit der ID immer bei Gebrauch der Tabelle angezeigt.

6.2.6. Dokumentation

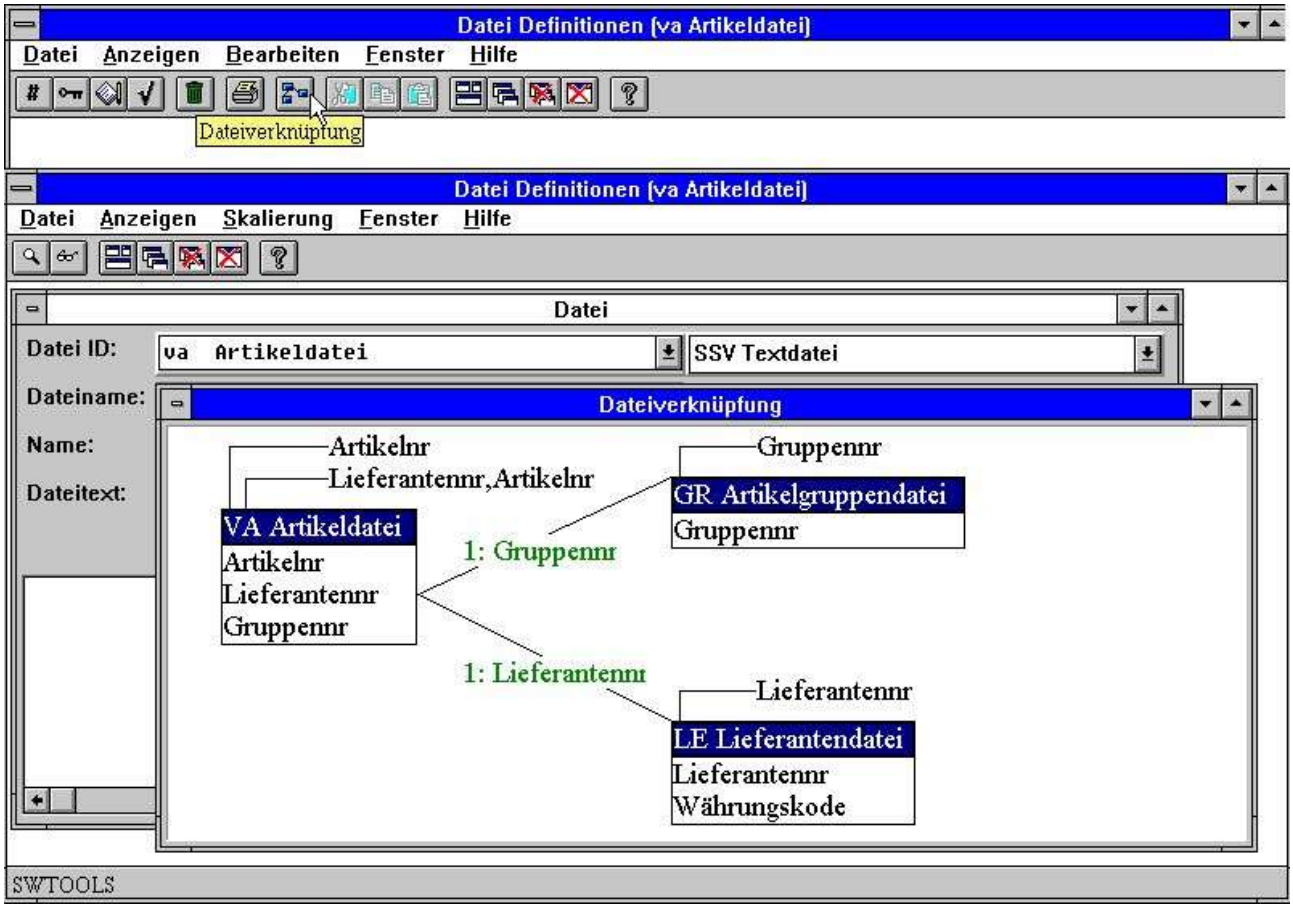
Die Tabelle kann hier in einem freien Text beschrieben werden. Dieser Text wird mit der Tabellendokumentation ausgeschrieben.

6.2.7. Standard ID

Die Standard ID identifiziert die Tabelle. Die SID wird im Zusammenhang mit PUT und LOAD von Standard Tabellenverknüpfungen (siehe oben) verwendet. Sollen diese Funktionen nicht benutzt werden, ist eine Angabe der SID nicht notwendig.
Beispiel für eine SID: AF-0500002031AS, wobei AF-05000 eine COMET-Datei mit dem Satztyp 020, COMET Version 3.1 und Master Datei-ID AS ist.

6.3. Anfrage auf Databankverknüpfungen

Jede Tabelle kann mit Verbindungen zu anderen Tabellen definiert werden. Bestehen solche Verbindungen, können diese graphisch angezeigt bzw. ausgedruckt werden. Hierfür wählen Sie:

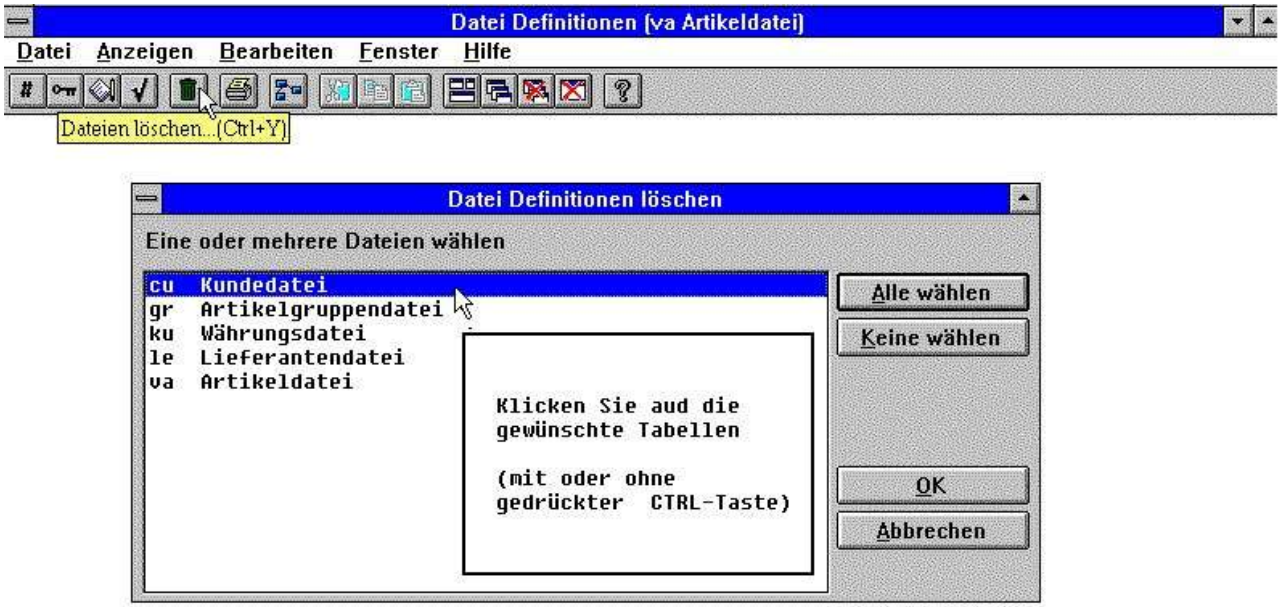


17. Datenbankverknüpfungen

Skalierung und Generierung mehrerer Niveaus ist möglich.

6.4. Löschen von Tabellen

Man kann definierte Tabellen mit Hilfe folgender Funktion löschen:



18. Löschen von Tabellen

6.5. Überblick über Datensätze

Diese Funktion gibt Ihnen einen Überblick über die ersten 100 Datensätze in einer Tabelle. Werden Datensätze einer BASIC Datei angezeigt, wird auch der Kontrollsatz im FDF Modul angezeigt.

19. Satzüberblick

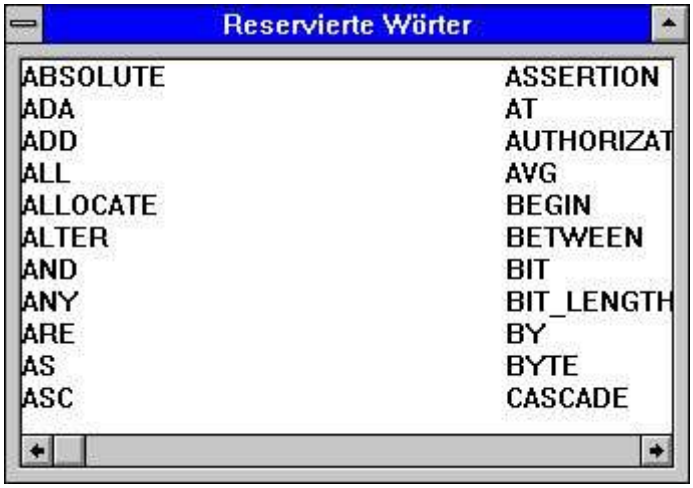
Wird unter dieser Funktion nichts angezeigt, kann dies folgende Ursachen haben.

- **Verkehrter Treibertyp**
- **Falscher Tabellenname**
- **Keine Sätze in der Tabelle**

Ein Unterfenster zeigt eventuelle Fehlermitteilungen beim Zugriff auf Daten mit: OPEN und READ an.

6.6. Reservierte Worte

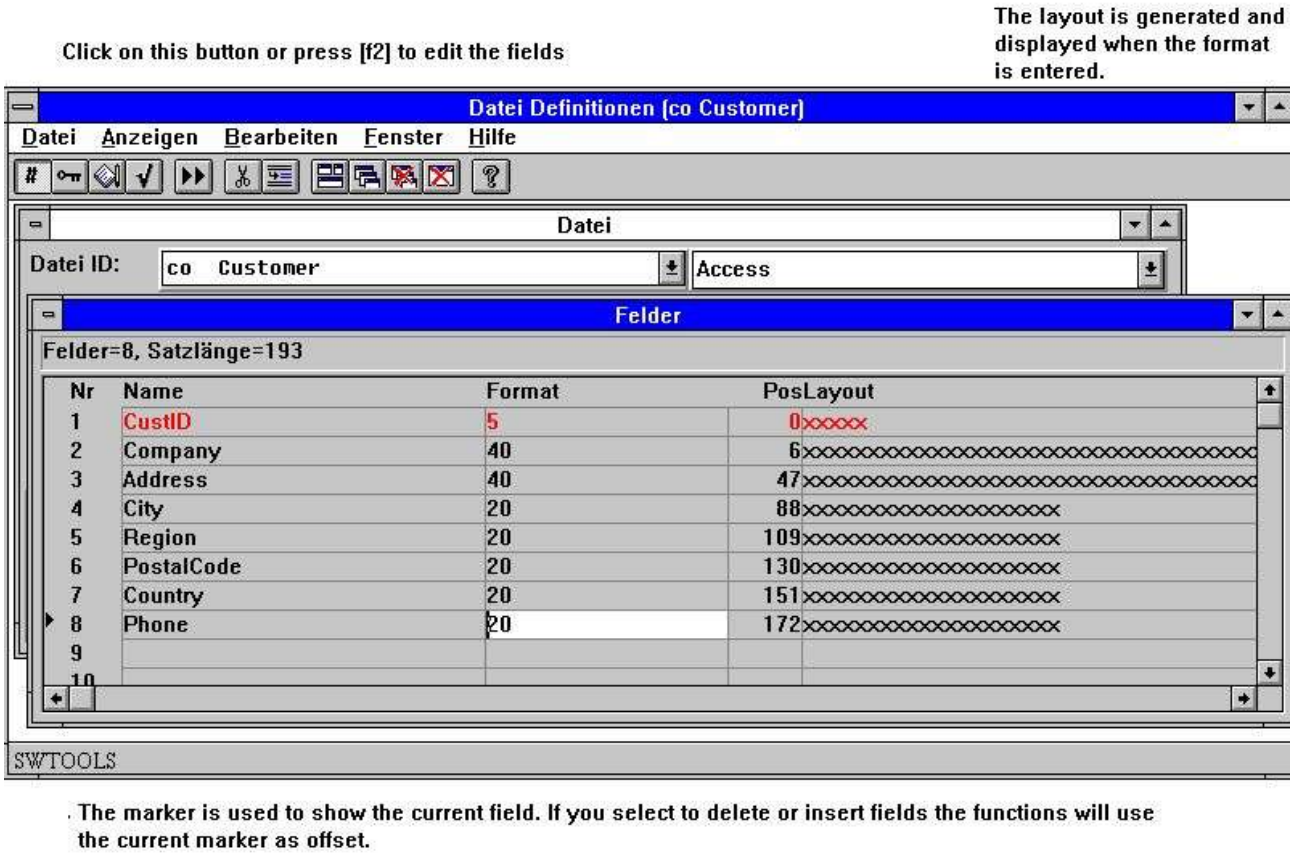
Diese Funktion gibt Ihnen einen Überblick über die in ODBC reservierten Worte. Dies kann bei Eingabe von Tabellennamen u.ä. von Nutzen sein.



20. Reservierte Worte

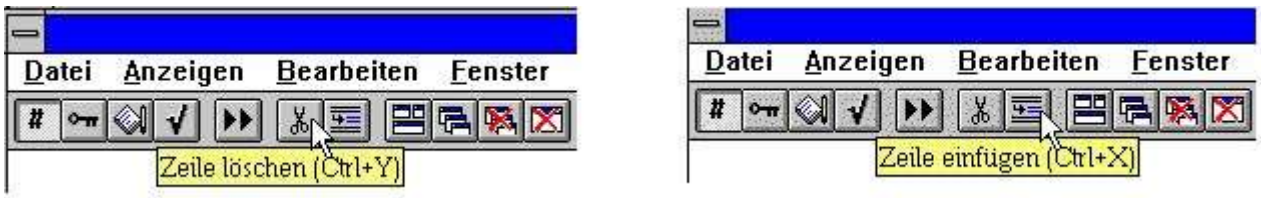
7. Feldbeschreibungen

Ein Feld ist eine Beschreibung wie ein bestimmter Wert in der Tabelle gelesen werden soll. Das Feld ist mit einer Nummer, einem Namen und dem Format definiert. Die Nummer wird automatisch zugeordnet. Der Name kann eingegeben werden, und wird in allen Feldübersichten benutzt. Das Format gibt an, wie ein Wert abgespeichert ist, und wie dieser in SW-Tools-TRIO angezeigt bzw. ausgedruckt werden soll.



21. Feldbeschreibungen

Felder können mit Hilfe der folgenden Funktionen eingefügt bzw. gelöscht werden:



22. Einfügen und Löschen von Feldern

Beim Einfügen bzw. Löschen von Feldern sollten Sie beachten, daß Ihre IQ-Programme und Listen, die Feldnummernverweise benutzen, hiervon beeinflußt werden.

7.1. Feldnummer

Die Feldnummer wird automatisch vom System zugeordnet, und kann nicht eingegeben werden.

7.2. Feldname

Der Feldname kann Buchstaben, Ziffern, Leerstellen und Sonderzeichen enthalten. Der Feldname wird normalerweise als Überschrift in Listen und Anfrageprogrammen verwendet.

7.3. Format

Das Feldformat gibt ein logisches Format an. Hierüber wird definiert, wie die Daten gespeichert und später gelesen bzw. angezeigt/gedruckt werden sollen. Das Format kann folgende Typen definieren:

- ein alphanumerisches Feld (Text)
- ein numerisches Feld (Ziffern)
- ein Datumsfeld

Ein alphanumerisches Feld ist nur mit seiner Länge beschrieben, z.B.:

10	Das Feld ist begrenzt auf 10 Stellen
20	Das Feld ist begrenzt auf 20 Stellen

Ein numerisches Feld kann nur numerische Werte enthalten, und kann mit oder ohne Dezimalstellen definiert werden. Die folgenden Beispiele beschreiben die Syntax für die Definition eines numerischen Formates:

2,	Wert von 0 bis 99 (zweistellig)
-2,	Wert von -99 bis +99 (zweistellig, mit Vorzeichen)
5,	Wert von 0 bis 99999 (fünfstellig)
-7,2	Sieben Stellen plus zwei Dezimalstellen, mit Vorzeichen
9,3	Neun Stellen plus drei Dezimalstellen (ohne Vorzeichen)

Das Datumfeld kann wie folgt definiert sein:

,6,	YYMMDD, wobei YY=Jahr, MM=Monat, DD=Tag
,8,	YYYYMMDD, wobei YYYY=Jahr, MM=Monat, DD=Tag

7.3.1. Ausgabe numerischer Felder

Der Wert eines numerischen Feldes kann in verschiedener Weise ausgegeben werden. Tausender Trennung erreicht man durch vorangestelltes Komma, z.B.

,9,3
-123,456,789.123

Ein Feld mit vorangestelltem Komma unterstützt gleichzeitig negative Werte. Die Tausender Trennung durch Punkt oder Komma wird im Präferenz Menü festgelegt.
Felder mit dem Format ,6, und ,8, muß als ,6,0 und ,8,0 eingegeben werden.
Vorangestellte Nullen werden durch ein & in der Formatangabe definiert:

9,3&
000012345.123

Vorangestellte * für Betragsfelder werden durch ein * in der Formatangabe definiert:

9,3*
******12345.123**

Das Vorzeichen kann voran- bzw. nachgestellt definiert werden:

9,2-
12345.12-

Nullenunterdrückung wird durch ein ' in der Formatangabe festgelegt:

9,2'

7.3.2. Fließendes Dezimalkomma

Mit der Angabe 9,5? im Feldformat kann ein Feld definiert werden, daß bei der Ausgabe 0 bis 5 Dezimalstellen angibt, abhängig vom Feldinhalt.
Zusammen mit dem Feldformat stehen eine Reihe von Optionen für die Definition der Datensätze zur Verfügung. Entsprechend den verschiedenen Datenbanksystemen und Dateitypen, die Data-Dictionary unterstützt, können auch unterschiedliche Feldarten verwendet werden.

7.3.3. (nnn) Tabellenfelder

2(010)

definiert ein zweistelliges alphanumerisches Feld, das 11 mal, numeriert von 0 bis 10, auftritt. Diese Felder werden unmittelbar nacheinander im Satz angelegt. Diese Felder können in Berechnungen als z.B. #7(0),#7(1),...,#7(11) benutzt werden.

7.3.3.1. (nnn+) Feldgruppen in Tabellen

2(010+)
4,(010+)
8,(010)

definiert eine Gruppe von Feldern, die jeweils 11 mal auftritt.

7.3.4. W Arbeitsfelder

10W

definiert eine Feld als Arbeitsfeld. Dieses Feld wird nicht mit einem READ gelesen, sondern wird für die Abspeicherung eines Schlüsselwertes, der nicht selbst im Datensatz gespeichert ist, verwendet.

7.3.5. (nn) Explizite Satzlänge

4(16)

definiert ein alphanumerisches Feld für 4 Zeichen, das aber mit 16 Stellen gespeichert wird. Es werden also 8 Bytes für eventuelle spätere Änderungen, typisch Erweiterung eines Schlüsselbegriffes, reserviert.

7.3.5.1. (nn,x) Explizite Satzlänge in Bits

4(16,4)

Definiert ein Feld, das 16 Bytes und 4 Bits belegt.

7.3.6. Bnn Explizite Bytenummer

8,B100

definiert ein Feld, das mit dem Byte Nr. 100 beginnt. Die Bytes sind durchnumerieret von 0 aufwärts. Beachten Sie bitte, daß in einigen Datenbanksystemen Felder auf bestimmten Bytegrenzen beginnen, z.B. beginnen in BASIC alle Felder auf Wortgrenzen, d.h. jeweils 16 Bit.

7.3.6.1. Bnnn,x Explizite Bitnummer

8,B45,4

definiert ein Feld, das mit Byte 45, Bit 4 beginnt.

7.3.7. Tn Fester Feldtyp

4,T1

definiert ein Feld mit 4 Stellen des Typs 1 in dem entsprechenden Datanbanksystem. Z.B. in einem BASIC-System, wo 4 Stellen in einer Variablen mit der Länge 1 Wort, das auf den Wert 7999 begrenzt ist, gespeichert werden.

7.3.8. Unn Spezifikke Kodetabelle für ein einzelnes Feld

32U2

Für ein einzelnes Feld kann eine Kodetabelle (Nummer) angegeben werden, die vorrangig vor der Kodetabelle für das System verwendet wird.

7.3.9. R Rechtsbündige Anzeige

10R

Ein alphanumerisches Feld wird rechtsbündig ausgegeben. Dies gilt auch für Schlüsselfelder (Indizes).

7.3.10. S Stopzeichen (delimiter) im alphanumerischen Feld

12S

Bewirkt die Teilung eines READ-Befehls in einem BASIC-Programm. Ist ohne Bedeutung in SW-Tools-TRIO Programmen.

7.3.11. Fnn Systemfeldadresse

Diese Option gilt nur für die Systemdatei. Hiermit werden Felder wie z.B. #RECNO mit einer bestimmten Datei verknüpft.

7.3.12. K,D - DATAMASTER Indexfelder

4K

definiert einen eindeutigen Schlüssel in TRIO DATAMASTER Dateien. Wird in anderen Dateien ignoriert.

4D

definiert einen mehrdeutigen Schlüssel in TRIO DATAMASTER Dateien. Wird in anderen Dateien ignoriert.

7.3.12.1. E,I,V,X,Z - DATAMASTER Spezial Indexfelder

Diese Option ist für eine Kompatibilität mit dem BASIC DATAMASTER System reserviert. Wird zur Zeit nicht verwendet.

7.3.13. Cnn - DATAMASTER Feldprüfung

Diese Option ist für eine Kompatibilität mit dem BASIC DATAMASTER System reserviert. Wird zur Zeit nicht verwendet.

7.3.14. Qnnn Explizit SQL Typ

19Q11

definiert a TIMESTAMP Feld für den ODBC Zugriff auf dieses Feld. Der SQL Typ wird normalerweise durch die Importdefinitionen bestimmt, doch gilt speziell

9 DATE
10 TIME
11 TIMESTAMP

7.3.15. Zugriffs Modus

A0 ermöglicht Lesen und Schreiben (read/write), A1 nur Lesen.
Diese Option wird in ODBC SQL Befehlen benutzt, wo A1 verwendet werden kann, um automatische Zählfelder von UPDATE Befehlen auszuschließen.

7.3.16. Pnn Gepackte Felder

Diese Option bestimmt, wie ein Feld in dem Datenbanksatz gespeichert wird. Dies ist in hohem Grade abhängig von dem aktuellen Datenbanksystem.

8,P

definiert ein gepacktes numerisches Feld mit 8 Stellen. Wie jedoch der Wert gepackt wird, kann nur mit Kenntnis des jeweiligen Datenbanktreibers bestimmt werden.
4P2 definiert ein alphanumerisches Feld mit dem Packungstyp 2 des jeweiligen Datenbanktreibers. Es wird hier auf die Beschreibung des entsprechenden Treibers verwiesen.

7.3.16.1. Pnnnn Explizite Angabe des Packtypes

Die Packtypen 0-999 sind für den Gebrauch individueller Datenbanktreiber reserviert. Die Typen 1000-9999 gelten für den jeweils installierten Treiber, unabhängig von der Art des Treibers.

,6,P1040

definiert immer eine ODBC DATUM Struktur, auch wenn dieses Feld zufällig in einer BASIC-Datei auftritt.

Typ	Name	Gespeichert als	Anwendung
0-999		Treiber abhängig	
1000	LDCHAR	Alphanumerish mit nachfolgende Blanke	C-Isam
1001	LDINT	2-bytes integer	C-Isam
1002	LDLONG	4-bytes integer	C-Isam
1003	LDFLOAT	4-bytes float	C-Isam
1004	LDDBL	8-bytes double	C-Isam
1005	SHORT	C-variabel short	C
1006	LONG	C-variabel long	C
1007	FLOAT	C-variabel float	C
1008	DOUBLE	C-variabel double	C
1009	FLEXBCD	BCD	Dataflex
1010	FLEXBCDDATE	BCD-date	Dataflex
1011	CHAR	C-variabel char	C
1012	UCHAR	C-variabel unsigned char	C
1013	SCHAR	C-variabel signed char	C
1014	NUMCHAR	Nicht implementiert	C
1015	ULONG	C-variabel unsigned long	C
1016	UINT	C-variabel unsigned short	C
1020	BTINT1	Integer 1-byte	BTrieve
1021	BTFLOAT	Float 4-bytes	BTrieve
1022	BTDATE	Datum	BTrieve
1023	BTTIME	Zeitpunkt	BTrieve
1024	BTDECIMAL	Decimal	BTrieve
1025	BTLOGICAL	Nicht implementiert	BTrieve
1027	BTNUMERIC	Numeric	BTrieve
1028	BTBFLOAT	BFloat	BTrieve
1029	BTLSTRING	LString	BTrieve
1030	BTZSTRING	ZString	BTrieve
1031	BTUBINARY	Nicht implementiert	BTrieve
1032	BTAUTOINC	Nicht implementiert	BTrieve
1033	BTINT2	Integer 2-byte	BTrieve
1034	BTINT4	Integer 4-byte	BTrieve
1035	CONNUM	Varlänge double Betrag	Concorde
1036	CONNUM4	Varlänge long	Concorde
1037	CONALF	Varlänge text	Concorde
1038	CONDATE	Datum	Concorde
1039	CONNUM2	Unsigned int	Concorde
1040	ODBCDATE	Datum Struktur	ODBC
1041	ODBCTIME	Zeitpunkt Struktur	ODBC
1042	ODBCSTAMP	Timestamp Struktur	ODBC
1043	RCHAR	Rechtsbündige Text	Concorde
1044	CONDBL	Feste länge double Betrag	Concorde 4
1045	CONDAF	Feste länge datum	Concorde 4
1046	RMIDBL	RM double	C-RM

1047	STRDATE	Datum JJJJMMTT als Text	C-RM
1048	ODBCSL1	Text minimum 1 stelle	Access
1049	LDINTI	Als LDINT, aber bytes rundgekehrt	CX-Basic
1050	LDLONGI	Als LDLONG,aber bytes rundgekehrt	CX-Basic
1100	PARITY	Alphanumerish mit parität	Basic $
1101	BCD1	1-word BCD integer	Basic 1%
1102	BCD2	2-word floating point Nummer	Basic 2%
1103	BCD3	3-word floating point Nummer	Basic 3%
1104	BCD4	4-word floating point Nummer	Basic 4%
1105	BCD5	5-word floating point Nummer	UNIBASIC 5%
1106	CALL60	Wert gepacked mit CALL 60	Basic gepacked
1107	NX2	Wert gepacked mit CALL 60, keine Nachkommastellen	Basic
1108	NX3	Feld alphanumerish gespeichert mit Nachkommastellen	Basic
1109	NX4	Feld alphanumerish gespeichert ohne Nachkommastellen	Basic
1110	NX5	Gespeichert als 1%, keine Nachkommastellen	Basic
1111	BINARY	Binary gespeichert	Cobol comp-3
1112	GSM1	Speedbase Alpha (x bytes)	Global
1113	GSM2	Speedbase 1-2 Ciffern numerish (1 byte)	Global
1114	GSM3	Speedbase 3-4 Ciffern numerish (2 bytes)	Global
1115	GSM4	Speedbase 5-6 Ciffern numerish (3 bytes)	Global
1116	GSM5	Speedbase 7-9 Ciffern numerish (4 bytes)	Global
1117	GSM6	Speedbase >9 Ciffern oder (Cif+Dec)/2 bytes	Global
1118	GSM7	Speedbase datum Feld mit fna()/fnb()	Global
1201-19	CROSS	BCD: S | exp | bcd 01-19 Ciffern	C-Isam
1221-39	CROSS	BCD: bcd 01-19 Ciffern | exp | S	C-Isam
1241-59	CROSS	BCD: bcd 01-19 Ciffern	C-Isam
1053	LDCHARR	Rechtsbündig, kein extra Byte wie 1043	C-Isam
1121	NAVI_STR	Textfolge	Navision P5
1122	NAVI_DATE	Datum	Navision P6
1123	NAVI_TIME	Uhrzeit	Navision P3
1124	NAVI_BLOB	Binäres großes Objekt	Navision
1125	NAVI_BOOL	Boolean	Navision P2
1126	NAVI_S16	Kurz	Navision P7
1127	NAVI_S32	Lang	Navision P8
1128	NAVI_U8	Zeichen (byte)	Navision P1
1129	NAVI_ALFA	Textfolge mit 1.byte=Länge	Navision P
1130	NAVI_BCD	Betrag	Navision P9
1131	NAVI_O32	Optionskode	Navision P4

7.3.17. Vxxx Sekundäre Packungstypen

Im Zusammenhang mit den Jahr 2000 Erweiterungen wurde es notwendig, einen sekundären Packungstyp einzuführen. Dieser Packungstyp kann so verstanden werden, daß eine festgelegte Berechnung nach Lesen und Entpackung des Feldes entspr. Packungstyp Pxxxx, bzw. vor Packung und Rückschreiben,vorgenommen wird.
Der sekundäre Packungstyp Vxxx wird im Feldformat angegeben wie: ,6,V1

Typ	Name	Berechnung	(Standard) Parameter ZZ
V	JJMMTT	Kurzdatum JJMMTT -> JJJJMMTT	(50) Jahr 2000 wenn JJ<=ZZ
V1	TTMMJJ	Kurzdatum TTMMJJ -> JJJJMMTT	(50) Jahr 2000 wenn JJ<=ZZ
V2	TTTTTT	Tagesnr. TTTTTT -> JJJJMMTT	(0) Basistag V2.BASENUMBER
V3	NXU2000	Basic ungepackt ZJmmdd, Z=?@ABC --> 00-09	Keine
V4	NXP2000	Basic gepackt ZJmmdd, Z=Leerst.+,-. --> 0-4	Keine
V8	OCTAL	Konvertierung von oktal bei Lesen, "1234" -> 668	Keine
V16	HEXA	Konvertierung von hex bei Lesen, "abcd" -> 43981	Keine

7.3.17.1. Parameter für sekundären Packungstyp

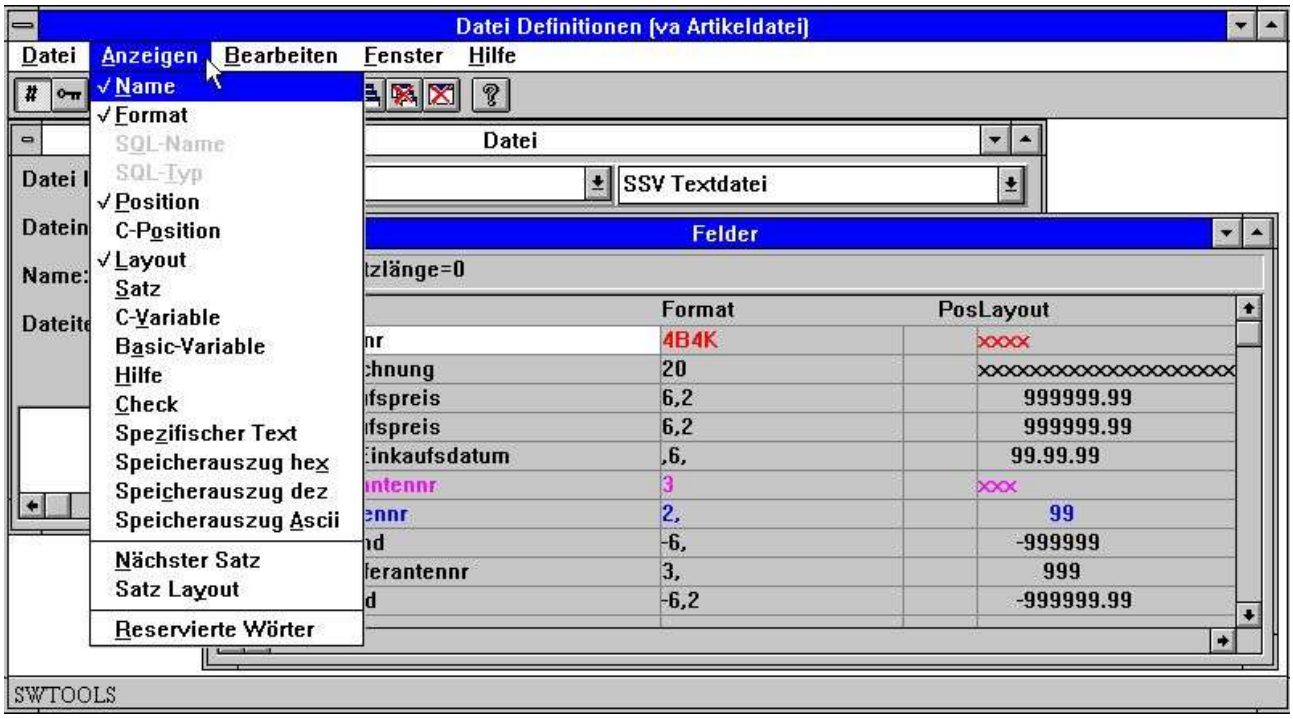
Zusammen mit dem sekundären Packungstyp kann ein Parameter angegeben werden, z.B. ,6,V1.30.
Hiermit kann die Standardkonvertierung von/zu Kurzdatum Formaten außer Kraft gesetzt werden.

7.3.17.2. Standard für sekundäre Packung für BASIC Datum

Für Kurzdatumfelder (,6,) in BASIC Dateien ist die sekundäre Standardpackung in numerischen Feldern V, in alphanumerischen V3, und V4 in gepackten Feldern.
Diese Regeln gelten auch für Schlüsselfelder, die ein Kurzdatum enthalten.

7.4. Informationen in der Feldzeile, VIEW-Menü

Standardmäßig werden Bytenummer und Layout des Feldes in der Feldzeile angezeigt. Über das VIEW Menü kann der Inhalt der Feldzeile geändert werden:



23. VIEW Menü

7.4.1. Bytenummer

Die Bytenummer wird automatisch berechnet und bei Gebrauch angezeigt. Die Bnnn-Option kann im Feldformat verwendet werden, wenn ein bestimmtes Byte angegeben werden soll, z.B. wenn nicht alle Felder definiert sind oder die gewünschte Reihenfolge nicht der Reihenfolge im Datensatz entspricht.

7.4.2. Feldlayout

Das Feldlayout zeigt, wie das Feld gedruckt bzw. am Bildschirm angezeigt wird. Bei numerischen Feldern wird die maximale Feldlänge durch eine Anzahl von X angegeben. Bei alphanumerischen Feldern geschieht dies durch eine Anzahl von 9.

7.4.3. SQL-Namen

Die SQL-Namen können nur in ODBC-Tabellen angesprochen werden. Der SQL-Name ermöglicht den Gebrauch von Namen für Überschriften, die unterschiedlich zum SQL-Namen, der von dem Datenbanktreiber benutzt wird, sind.
Der SQL-Name wird direkt an den ODBC-Treiber, falls dieser vorhanden ist, übergeben. Dies bedeutet, daß die Syntax, die vom aktuellen Treiber unterstützt wird, eingehalten werden muß. Namen sollten deshalb keine Leerstellen oder Sonderzeichen enthalten, es sei dann, diese sind in '...' eingeschlossen.
Im EDIT-Menü können SQL-Namen aus den bestehenden Feldnamen generiert werden, wobei die Standardregeln eingehalten werden. Sie können auch selbst diese Namen eingeben, wobei der Gebrauch der besonderen Spalten COUNT(*) und A+B zugelassen ist.

Nr	Name	Format	SQLname	SQLtype
1	ArtikelNr	-5,	ProdID	LONG
2	Name	40	ProdName	TEXT(x)
3	Preis	-18,4	UnitPrice	CURRENCY
4	Meine eigene Text	64	Descript	TEXT(x)
5	Gewicht	-15,	ShipWt	CURRENCY

24. SQL-Namen und SQL-Typen

7.4.4. SQL-Typen

Auch die SQL-Typen sind nur in ODBC Tabellen ansprechbar. Die angezeigten Typen sind Standardtypen entsprechend den Feldformaten, und sind abhängig von der gewählten ODBC Datenbank. Werte in dieser Spalte können nicht geändert werden. Hierzu muß die Qnnn-Option im Feldformat verwendet werden.

7.4.5. C-byte

Das C-Byte wird von dem internen Programmsatz benutzt und zeigt an, wie die Daten in der C-Struktur gespeichert sind. Diese Information ist nur für Entwickler von Bedeutung.

Nr	Name	Format	Pos	C-Pos	Cvar	Basicvar
1	Satzart	2,	0	0	short	S1(1)
2	Nr	5SP	2	2	char	S$(1,5)
3	Name	30B8	8	8	char	S$(3,32)
4	Name 2	30	38	39	char	S$(33,62)
5	Straße	30	68	70	char	S$(63,92)
6	Ort	30	98	101	char	S$(93,122)
7	Land	30	128	132	char	S$(123,152)
8	Postleitzahl	11	158	163	char	S$(153,163)
9	FW-Code, F/V-Code	2,B170	170	175	short	S1(2)
10	Brutto-Netto-Kennzeichen	1,	172	177	short	S1(3)

25. Byte, C-Byte, C-Variable und BASIC Variable

7.4.6. C-Variable

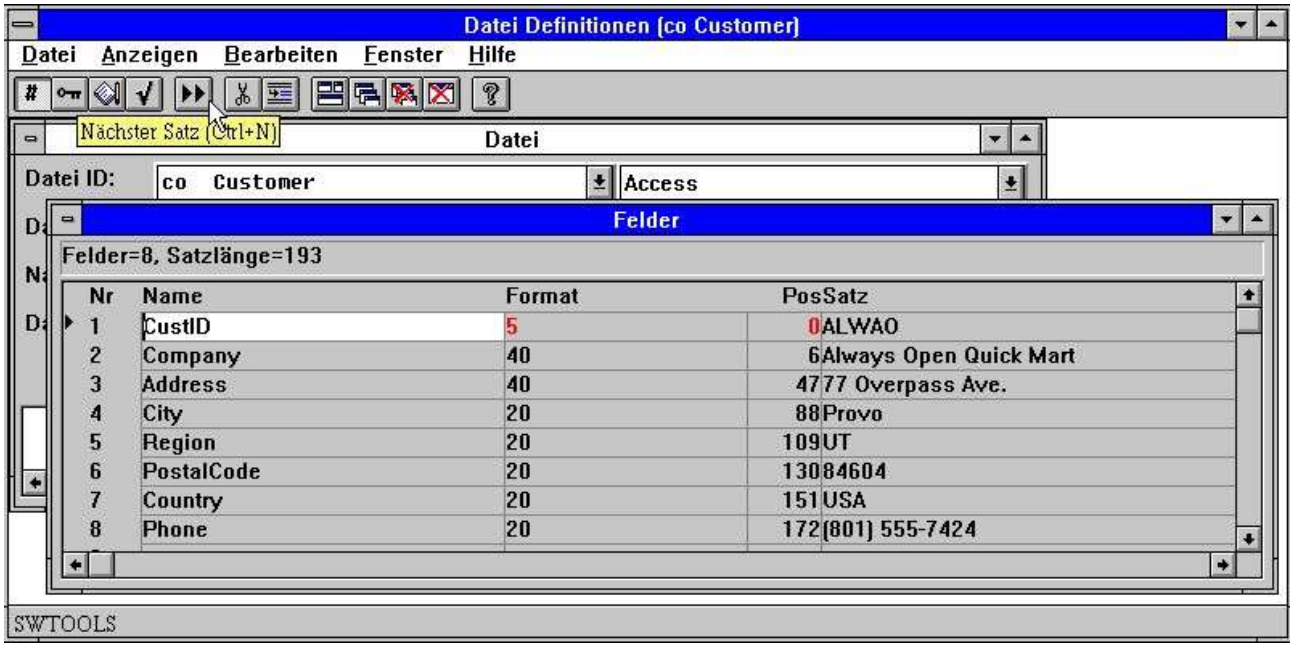
Die Option ist nur für Entwickler von Bedeutung. Es wird die C-Variablentype, die intern benutzt wird (char, short, long, float, double), gezeigt.

7.4.7. BASIC-Variable

Beim Entwickeln von BASIC Programmen kann der Zugriff aus die BASIC Variable von Vorteil sein. Für C Programme (wie TRIO) hat diese keine Bedeutung.

7.4.8. Datensatz

Falls die Tabelle zugänglich ist (kann eröffnet werden), können Satzinhalt zusammen mit den Felddefinitionen angezeigt werden. Immer wenn eine Felddefinition geändert wird, wird dies in der entsprechenden Zeile sofort angezeigt. Hierdurch erhält man die Möglichkeit, Änderungen unmittelbar zu prüfen.



26. Satzinhalt einer Tabelle

Die Anzeige des Satzinhaltes kann über dieses Menü ein-/ausgeschaltet werden.

7.4.9. Nächster Satz

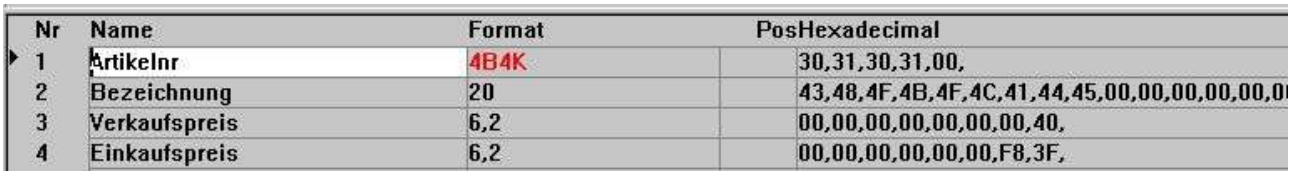
Diese Funktion liest den nächsten Satz in der Tabelle. Kann die Tabelle z.B. aufgrund eines nicht zulässigen Namens oder einem verkehrten Treibertyp nicht geöffnet werden, werden keine Werte angezeigt.
Diese Funktion ist auch über die Symbolleiste verfügbar. Die Funktion 'nächster Tabellensatz' aktiviert automatisch die Option 'Anzeige des Tabellensatzes'.
Das erste mal, wenn die Funktion aufgerufen wird, wird der erste Satz in der Tabelle gelesen, anschließend der nächste uzw. bis zum Tabellenende.

7.4.10. Satzüberblick

Diese Funktion zeigt die ersten 100 Sätze am Bildschirm an (siehe auch 'Tabelleninformation').

7.4.11. Hexadezimaler Dump eines Satzes

Diese Funktion schreibt die Tabelle hexadezimal am Bildschirm aus.



Nr	Name	Format	PosHexadecimal
1	ArtikelNr	4B4K	30,31,30,31,00,
2	Bezeichnung	20	43,48,4F,4B,4F,4C,41,44,45,00,00,00,00,00,0
3	Verkaufspreis	6,2	00,00,00,00,00,00,00,40,
4	Einkaufspreis	6,2	00,00,00,00,00,00,F8,3F,

27. Hexadecimaler Dump eines Tabellensatzes

Diese Funktion dient dem Programmierer um evt. Probleme lösen zu können.

7.4.12. Dezimaler Dump eines Satzes

Falls gewünscht, kann der Dump auch in dezimaler Notation erfolgen.

7.4.13. Ascii Dump eines Satzes

Der Tabellensatz kann auch in Textform angezeigt werden.

7.4.14. Übersicht über Hilfstexte

Diese Option zeigt die erste Zeile eines evt. Hilfstextes für ein Feld an.

9	FW-Code, F/V-Code	2,B170		FB134-137
10	Brutto-Netto-Kennzeichen	1,		0 =Brutto
11	WORD-Kennzeichen	2,		0 =keine Integration
12	Skontogruppe	2,		0-40
13	Rechnungslegungstag	2,		
14	Zahlungziel	4,T1		MAX 7999

28. Übersicht über Hilfstexte

7.4.15. Übersicht über Prüfvorschriften

Falls Prüfvorschriften definiert sind, werden diese zusammen mit den Felddefinitionen mit dieser Funktion angezeigt.

7.4.16. Übersicht über Sondertexte

Sind Sondertexte für die Felder definiert, werden die jeweils ersten Zeilen dieser Texte angezeigt.

7.4.17. Reservierte Worte

Diese Funktion zeigt die reservierten SQL-Wörter an.

7.5. Erweiterung von DATAMASTER Dateien mit dem FDF Modul

Erweitern Sie eine DATAMASTER Datei mit dem FDF Modul, verbleibt die Datei selbst unverändert, wohingegen die Dateidefinitionen geändert werden. Ihnen wird folgende Frage gestellt:

29. Bestätigung einer DATAMASTER Dateikopie

7.5.1. DATAMASTER Datei Konvertierung von BASIC in ODBC

Wird eine BASIC Datei mittels ODBC mit dem DATAMASTER kopiert, werden die Informationen betr. Packtype, Bytenummer und Stopzeichen entfernt, da diese die empfangende ODBC Tabelle verfälschen können.

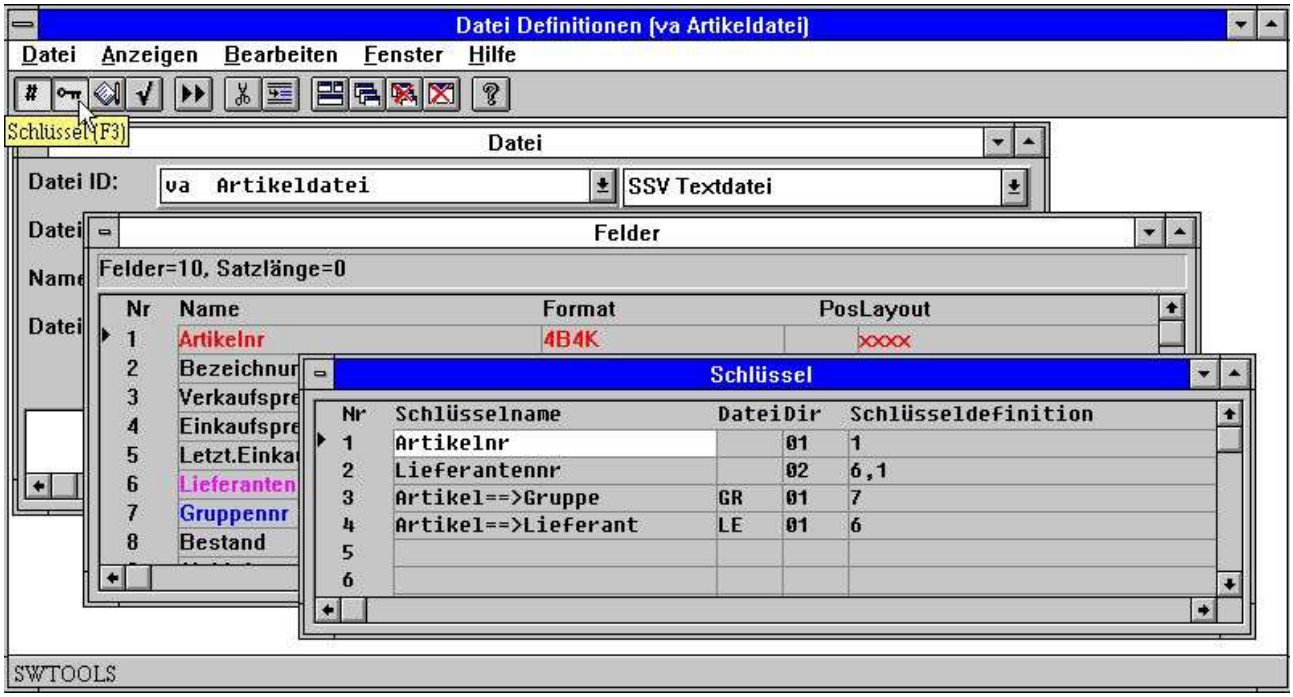
8. Index, Hilfstexte und Prüfvorschriften

Über das EDIT Menü können folgende Funktionsbereiche aufgerufen werden:

30. EDIT Menü

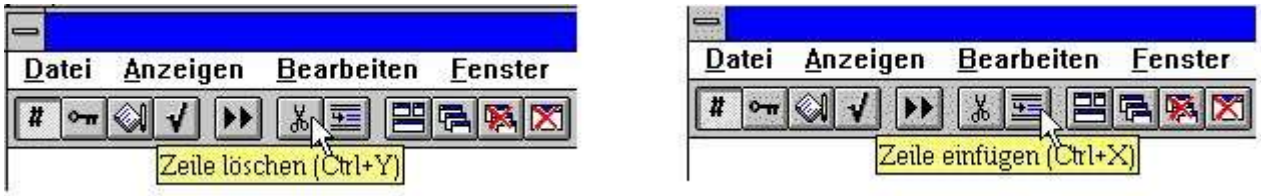
8.1. Index-Definitionen und Tabellenverknüpfungen

Die Indizes definieren, wie die Datensätze in einer Tabelle sortiert sind und wie auf einen gewünschten Satz zugegriffen werden soll.
Verknüpfungen zu anderen Tabellen definieren die gegenseitigen Verbindungen zwischen verschiedenen Tabellen.
Um die Indizes und Verknüpfungen für eine Tabelle auszugeben, ist folgende Funktion zu wählen:



31. Editierung der Indizes

Indizes können eingefügt und gelöscht werden, indem man die folgende Funktion wählt:



32. Einfügen und Löschen von Indizes

Sie haben sicher die unterschiedlichen Farben im Fenster bemerkt. De rote Farbe zeigt an, daß das Feld in einer oder mehreren Indexdefinitionen verwendet wird. Blau, daß das Feld für eine Verknüpfung benutzt wird, und Margenta, daß das Feld sowohl für Indizes, als auch für Verknüpfungen herangezogen wird.

8.1.1. Indexname

Der Indexname kann Buchstaben, Ziffern, Leerstellen und Sonderzeichen enthalten. Normalerweise wird der Indexname für Beschreibungen in SW-Tools TRIO bei Übersichten über die Tabellenindices verwendet, z.B. bei Start einer Listenausgabe.

8.1.2. Datei ID

Die Datei ID wird nur in Zusammenhang mit der Definition von Verbindungen zu anderen Dateien benutzt. Hierbei muß die verweisende ID einer im System bestehenden Datei ID entsprechen.
Im obigen Beispiel besitzt die Artikeldatei Verbindungen zu zwei weiteren Dateien. Zum ersten zu der Artikelgruppendatei, und zum zweiten zu der Artikellieferantendatei.

8.1.3. Directory

Das Directory, auch unter der Bezeichnung Indexnummer verwendet, ist eine zweistelliger Wert.
Bei Verknüpfungen zu anderen Tabellen definiert die Indexnummer, welcher Index in der verbundenen Datei bei einem Zugriff verwendet werden soll.
Im obigen Beispiel ist die Lieferantendatei LE definiert mit einem Primärindex, der Lieferantennummer. Die Lieferantennummer erhält man aus Feld 6 der Artikeldatei.
Die Indexnummer 00 ist in sofern speziell, als das hierüber Dateien angesprochen werden können, die Satznummern verwenden. Dies ist von Bedeutung, wenn eine Datei mit relativen Satznummern arbeitet, z.B. in einem BASIC System.

8.1.4. Schlüsseldefinitionen

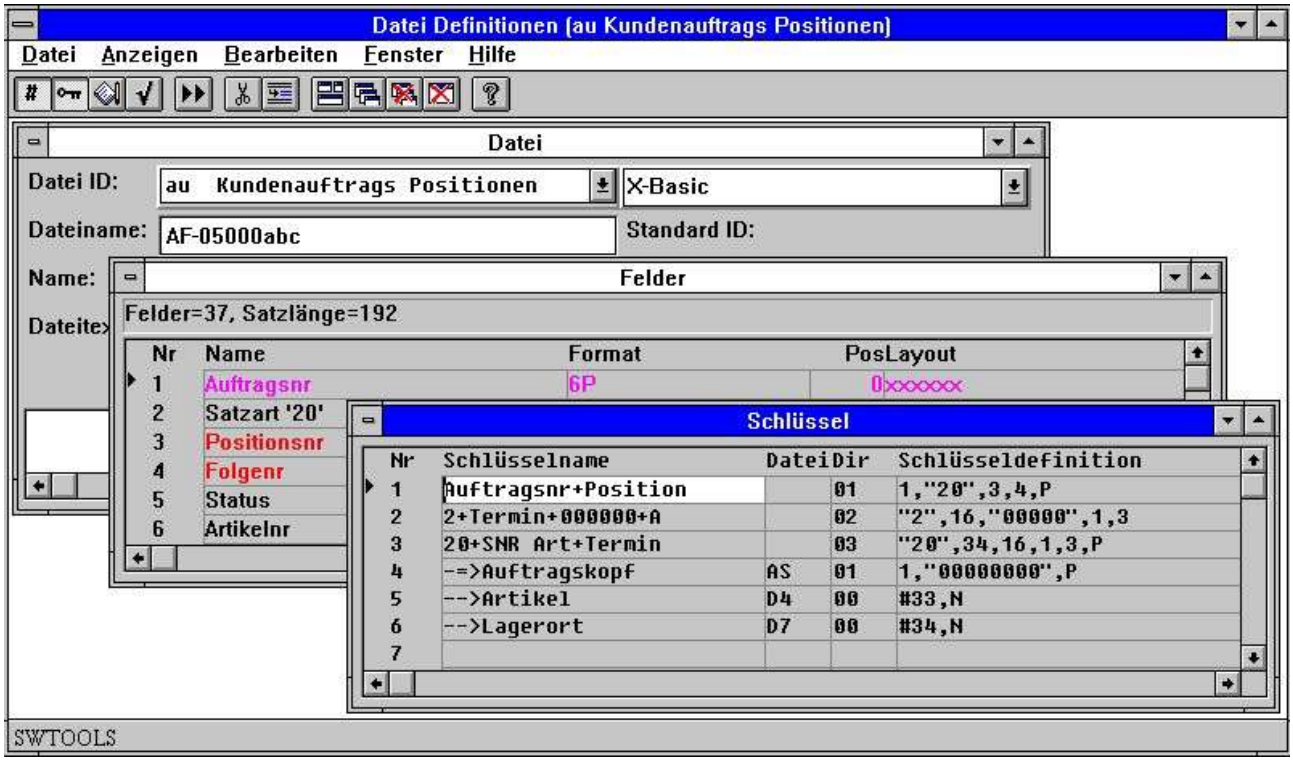
Eine Schlüsseldefinition ist die Definition eines Indizes oder einer Verknüpfung. Bei der Definition eines Indizes wird beschrieben, welches Feld(er) den Index ausmacht(en), und wie diese gespeichert sind.
Eine Schlüsseldefinition kann Verweise zu einer oder mehreren Feldnummern enthalten, kann aus einem Teil eines Feldes bestehen, und beschreibt, wie das Feld gepackt ist. Als Option kann ein # vor der Feldnummer angegeben werden.
Im obigen Beispiel definieren die ersten zwei Zeilen den primären und den sekundären Index für die Artikeldatei. Der Primäre Index verweist auf das Feld 1, Artikelnummer, und definiert, daß die Datei in aufsteigender Reihenfolge nach der Artikelnummer sortiert ist. Der sekundäre Index besteht aus zwei Feldern, Feld 6 (Lieferant) und Feld 1 (Artikelnummer).
Hiermit kann über die Lieferantennummer eine Liste, die nach Lieferant und innerhalb des Lieferanten nach Artikelnummern sortiert ist, aufgestellt werden.
Zeile 3 und 4 definieren die Verknüpfungen zu anderen Tabellen. Die Schlüsseldefinition verweist auf die Werte, die für den Zugriff auf verbundene Dateien notwendig sind. Der erste Verweis benutzt den Wert in Feld 7 (Artikelgruppennummer), um die Artikelgruppendatei über den Index 1 ansprechen zu können. Der zweite Verweis benutzt den Wert in Feld 6 (Lieferantennummer), um die Lieferantendatei über den Index 1 aufrufen zu können.

8.2. Index Betrachtungen

Oben ist eine einfache Schlüsseldefinition, die nur Felder und Kombinationen von Feldern beinhaltet, beschrieben. Jedes Element in einer Schlüsseldefinition wird als Teilschlüssel bezeichnet.
Die Anzahl der für eine Datei dafinierbaren Schlüssel, die Anzahl der Teilschlüssel, die Länge des Schlüssels und die Art des Packens sind ausschließlich von dem jeweiligen Treiber bestimmt. In diesem Zusammenhang verweisen wir auf die Beschreibungen der entsprechenden Treiber.
BASIC läßt max. 15 Schlüssel (Indizes) mit einer max. Länge von je 30 Bytes zu. Da der Schlüssel immer als Zeichenfolge betrachtet wird, ist die Anzahl der Teilschlüssel und der Gebrauch von Teilfeldern im Schlüssel nicht begrenzt.
Alle Schlüssel müssen eindeutig sein. Es ist nicht absolut notwendig, daß der Schlüssel selbst Teil des Datensatzes ist.
Isam Spezifikationen unterscheiden sich ein wenig, doch sind typisch bis zu 128 Indizes mit einer max. Länge von je 128 Bytes zugelassen. Jeder Schlüssel kann bis zu 8 Teilschlüssel enthalten. Ein Teilschlüssel muß immer ein gesamtes Feld repräsentieren, und muß Teil des Datensatzes sein.
ODBC SQL Datenbank Systeme benutzen Indizes, um die Verarbeitungszeit zu optimieren. Neue Schlüssel können jederzeit mit Hilfe von ORDRE BY eingerichtet werden. Beachten Sie jedoch bitte, daß hierdurch die Verarbeitungszeit beeinträchtigt werden kann.
Mit Hinblick auf die Unterstützung unterschiedlicher Datenbanksysteme, werden eine Reihe von Optionen im Zusammenhang mit der Schlüsseldefinition angeboten. Alle Optionen können benutzt werden. Es ist jedoch hierbei zu beachten, daß in Verbindung mit einigen Treibern gewisse Optionen wirkungslos sind bzw. zu Fehler auftreten können, Falls der Treiber die Optionen nicht unterstützt.

8.2.1. BASIC COMET AF-05000 Beispiel

Zuerst einen Blick auf eine Auftragszeile in BASIC COMET AF-05000:



33. AF-05000/020 Auftragszeile, Index und Verknüpfungen

8.2.2. Konstanten und Selektionen

1,"20",3,4,P

Es kann für den ersten Index eine Konstante, hier '20', angegeben werden. Der erste Index besteht also aufs dem Feld 1, der Konstanten 20 (Satztyp), und anschließend Feld 3 und 4.
Eine Konstante in einem Index bewirkt eine Selektion beim Lesen der Datensätze. Es werden also nur die Datensätze gelesen, wo der gelesene Wert mit der Konstanten übereinstimmt.
In COMET wird diese Möglichkeit typische verwendet, um bestimmte Satztypen zu selektieren, wenn unterschiedliche Satztypen in der gleichen Datei abgelegt sind. Jeder Satztyp sollte als separate logische Datei, mit Verknüpfung zu anderen Satztypen, definiert sein.

8.2.2.1. Mehrfach Konstanten in Indexdefinitionen

Dateien, die mit festen Satztypen im Index, wie "00",#1,P definiert wurden, können in z.B. "00,20-29,40",#1,P erweitert werden.
In einem Suchvorgang werden jetzt alle Sätze mit einer dieser Konstanten gesucht.

8.2.3. Packen

1,"20",3,4,P

Alls Teilschlüssel vor der Option ,P werden gepackt. Die P-Option kann auch für einzelne Felder benutzt werden, z.B. 1,"20",3P,4. In diesem Beispiel wird also nur das Feld 3 gepackt. Wie bereits erwähnt, erfolgt das Packen spezifik für jedes Datenbanksystem. Für BASIC muß z.B. der Aufruf CALL 60 verwendet werden.

8.2.4. Satznummern

#38,N

Auf die Satznummer der aktuellen Datei kann mit N, auf die relative Satznummer mit R verwiesen werden. Diese Option kann wie ein Feld für den Aufbau von Indizes bzw. für Verknüpfungen zu anderen Dateien benutzt werden.

8.2.4.1. Index 0

#38,N

Bei Index 0 wird die Datei unter Zuhilfenahme der Satznummer gelesen (nicht über einen Index). Bei Benutzung des Index = wird die Schlüsseldefinition wie eine Berechnung behandelt. D.h. daß # voran eines jeden Feldes stehen muß. Die Syntax wie #38+1+N ist zugelassen.

8.2.5. Duplikate zugelassen

#17,NP

Normalerweise muß der Schlüsselbegriff eindeutig sein. Bei Benutzung dieser Option sind Duplikate, z.B. mehrfach der gleiche Name, zugelassen.
In BASIC ist es gebräuchlich, die relative Satznummer (gepackt) dem doppelten Schlüssel folgenden zu lassen. Dies ergibt wiederum einen eindeutigen Schlüssel.

#17,M

In ISAM Systemen kann ,M für doppelte Indexwerte benutzt werden.

8.2.6. Nullen Unterdrückung

#17Z

Numerische Werte werden in Textfolgen konvertiert, um einen Schlüssel zu bilden. Wird die Option Z verwendet, werden vorangestellte Nullen in Leerstellen konvertiert.

8.2.7. Auffüllen der überzähligen Positionen

#17,S

S bewirkt, daß überzählige Positionen im Schlüssel mit Leerstellen aufgefüllt werden. Dies hat Bedeutung, wenn der Schlüsselbegriff länger als die entsprechende Textfolge ist.

8.2.8. Teilfelder

#17(2,4)

In diesem Beispiel werden die Positionen 2 bis 4 des Feldes 17 für den Schlüssel verwendet.

8.2.9. Schlüssel nicht Bestandteil des Satzes (KNIR)

In BASIC muß der Schlüssel nicht zwingend Bestandteil des Satzes sein. Der Aufbau der Schlüssel entspricht jedem anderen, jedoch müssen die Feldverweise mit einem W im Format gekennzeichnet sein. In diesem Falle werden die Indexwerte in die Felder während des Lesens der Datei eingesetzt. Diese Felder können wie normale Felder verarbeitet werden.
Beachten Sie jedoch, daß KNIR W Felder, die für den Index 1 definiert sind, nur gefüllt werden, wenn auch der Index 1 für den Satzzugriff verwendet wird. Wird der Index 2 benutzt, werden nur diese Felder gefüllt, nicht aber die Felder für Index 1.

8.2.10. Index Schlüsseldefinitionen, LOGICAL Index

Die Bedeutung von Index / Directory Nummern wurde überarbeitet.
Wird auf einen Index mit READ(xx.NN),z.B. beim Start einer Liste oder beim Lesen einer Datei zugeriffen, verweist man auf die ZEILENNUMMER in den Indexdefinitionen.
Die Zeilennummer wird jetzt zu einer LOGISCHEN INDEXNUMMER, die eventuell das physikalische Directory angibt, das in der Datei gespeichert ist.
Da die Indexzeilen früher entsprechend der Directorynummer der Datei sortiert wurden, gilt für alle bestehende Definitionen ZEILE=DIRECTORY. Es können also keine Probleme in der Rückwärtskompatibilität entstehen.
Diese Sortierung wurde entfernt, ausgenommen für DATAMASTER Dateien. Als Folge davon können Sie jetzt die Indizes in der von Ihnen gewünschten Folge definieren. Die Indexfolge wird z.B. im Zusammenhang mit Suchen in IQ Anfragen verwendet.
Die DIRECTORY NUMMER, die in einer Schlüsseldefinitionszeile angegeben ist, wird ausschließlich vom Datenbanktreiber für das physikalische Suchen in einem Directory benutzt. Es besteht kein Zusammenhang mit der INDEXNUMMER NN in READ(xx.NN).
Ein DIRECTORY kann mehrfach als LOGISCHE INDEXZEILE auftreten. Hiermit können unterschiedlich aufgebaute Suchschlüssel definiert werden. Dies wird ab und zu in BASIC Dateien benutzt, z.B. können unterschiedliche "XX" Konstanten als logische Indizes angegeben werden, ohne daß eine komplette Kopie der Dateidefinitionen nötig ist.

8.2.11. Directory 00 Satznummer lesen

Es kann auch im Index, verbunden mit dem Directory 00, also mit der Satznummer lesen, definiert werden. Dieser Index kann beim Start einer Liste, wie jeder andere Index, gewählt werden. Sie können auch ein KNIR Feld (Key Not In Record) bestimmen, das als Satznummer benutzt wird, wenn dieser Index zur Anwendung kommt.

8.2.12. ACCESS bei Verwendung von SWODBC in nichtindizierten Dateien

Microsoft ACCESS hat Probleme beim Zugriff auf nichtindizierte Dateien. Durch Definition des Index 00 wird dieses Problem übergangen.

8.2.13. Unterdrücken von IQ Suche bei speziellem Index (*xx)

Einer oder mehrere Indizes können bei einem IQ Suchvorgang unterdrückt werden. Hierfür muß die Directorynummer als *XX angegeben werden. Der Index wird übersprungen, falls man nicht gesondert angibt, daß der Suchvorgang vorgenommen werden soll.

8.2.14. Unterdrückung des Index-Update für einen speziellen Index

Die Angabe -XX als Directorynummer bewirkt, daß für BASIC Dateien kein Update entsprechend der Indexdefinition vorgenommen wird . Dies kann benutzt werden, wenn Mehrfachschlüssel für eine Datei definiert werden.

8.2.15. Anlegen eines Pseudoindexes (+xx)

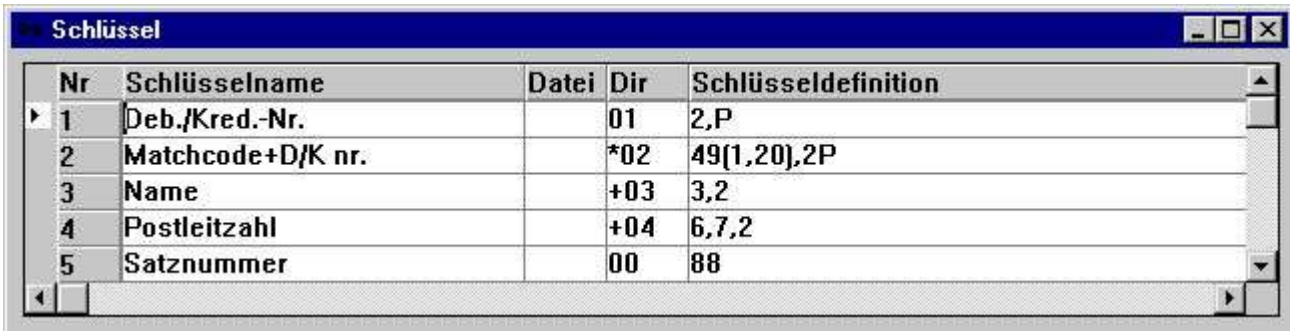
- DIESE FUNKTION IST Z.ZT. NICHT FREIGEGEBEN -

Die Angabe von +XX als Directorynummer, wobei XX ein nicht vorhandenes physikalisches Directory ist, bewirkt, daß intern für die erweiterte SSV Schnittstelle ein nutzbarer Index konstruiert wird. Setzen Sie XX-Zeilennummer immer so, daß der physikalische Index dem benutzten logischen Index entspricht.

Der Index kann benutzt werden, als sei er physikalisch vorhanden. Wird ein Server im System benutzt, wird der Index auf dem Server, nicht auf dem lokalen PC, konstruiert.

Aufgrund der Komplexität dieser Funktion wird die endgültige Freigabe erst nach weiteren intensiven Tests erfolgen. Die vorläufige Testfreigabe kann für Sie jedoch von Interesse sein.

8.2.16. Index Beispiel



34. Beispiel eines Pseudoindex und Satznummer Definition

8.2.17. Schlüsseloptionen, absteigender Index

Folgende Schlüsseloptionen wurden hinzugefügt:

I = Invertierung aller Bits im Schlüsselfeld (absteigender Index)
C = Nur GLOBAL Dateien, Unterdrückung des DMAM Index
S = Nur GLOBAL Dateien, keine Invertierung des VorzeichenBits in numerischen Feldern

Eine Indexdefinition wie z.B. 5I,2 invertiert alle Bits in Feld 5, also absteigender Index.

8.2.18. ODBC Zugriff, Beispiel

Beispiel einer Tabellendefinition für Auftragseingänge:

35. Zugriff auf Auftragseingänge, Tabelle mit SQL Namen

8.2.19. Index SQL-Name

In einer SQL-Datenbank hat jeder Index einen Namen, der bezüglich des Treibers gültig sein muß. Dieser Name wird nur bei der Anlage einer Tabelle bzw. bei der Neudefinition der Tabelle verwendet. Bei einem normalen Lesen muß dieser Name nicht angegeben werden.
Aus dem VIEW-Menü (während der Indexdefinitionen) kann die Spalte mit dem SQL-Namen für die Eingabe des Namens hinzugefügt werden. Bei Generierung von SQL-Namen wird ein Standard Name allen Indizes zugewiesen.

8.2.20. Natürliche Indizes und ORDER BY

#1,#2,#3,B

definiert einen Index, der aus den Feldern 1,2 und 3 besteht. Die Option B zeigt an, daß dies ein natürlicher Index ist, und daß ORDER BY nicht notwendig ist. Wird B ausgelassen, wird beim Lesen dem SELECT ein ORDER BY zugefügt.

8.2.21. Absteigende Indizes

#3,D

Mit Hilfe der Option D kann eine absteigende Indexfolge definiert werden.

8.2.22. Spezielle Kodetabelle

#2U2

U2 bewirkt, daß die Kodetabelle 2 benutzt werden soll, bevor ein Feld für den Index herangezogen wird.

8.3. Betrachten über Tabellenverknüpfungen

Tabellenverknüpfungen werden wie Indizes definiert.
Sie sollten jedoch beachten, daß eine Anwendung sowohl Verknüpfungsdefinitionen wie auch Indexdefinition zum Lesen einer Datei benutzt.
Eine Verknüpfung wird erreicht, indem man das Feld in der Indexdefinition mit den Feldern, die Dateiverknüpfungen definieren, füllt.

8.3.1. Abhängigkeit von Schlüssellänge

Dateitypen in Verbindung mit Abhängigkeit von Schlüssellänge #3(1,-2) wurden implementiert.

8.3.2. X* feste Verbindung zu Parameterdateien

Eine Parameterdatei, die immer einen bestimmten Satz lesen sollte, kann jetzt mit einer Verknüpfung definiert werden, die von allen READ(pa) benutzt wird, falls nicht anders angeben wurde.
Die Definition kann als Verknüpfung von PA zu Datei X* Index 0 Satznummer 5 angegeben werden.

8.3.3. Dateiverknüpfungs Definitionen

Wenn bei einem READ(xx.NN) mit NN auf die Zeilennummer des logischen Index verwiesen wird, geschieht dies auch bei der Definition von Dateiverknüpfungen in den Dateidefinitionen.
Für eine Verknüpfung muß in der Spalte DIRECTORY die Zeilennummer des logischen Index angegeben werden. Diese wird dann beim Lesen der Datei benutzt.

8.3.4. Mehrfach Verknüpfungen zur gleichen Datei mit Gross-/Kleinbuchstaben

READ(xx) sucht nach einer Verknüpfung zur Datei xx zuerst mit Kleinbuchstaben, und anschließend, falls nicht gefunden, auch mit Großbuchstaben. XX, Xx, xX, xx gibt also unterschiedliche Verknüpfungen an.
Die Sortierung von Dateiverknüpfungen wurde so geändert, daß alle Verknüpfungen zu einer Datei gemeinsam angezeigt werden (unabhängig Groß-/Kleinschreibung).

8.3.5. Mehrfach Verknüpfungen zur gleichen Datei durch logischen Index

READ(xx.02) benutzt den logischen Index 02 als Verknüpfung zur Datei xx. Dies kann auch dazu benutzt werden, mehrere Verbindungen mit unterschiedlichen Feldenr anzugeben.

8.3.6. Vorgegebene Verknüpfungen bei READ

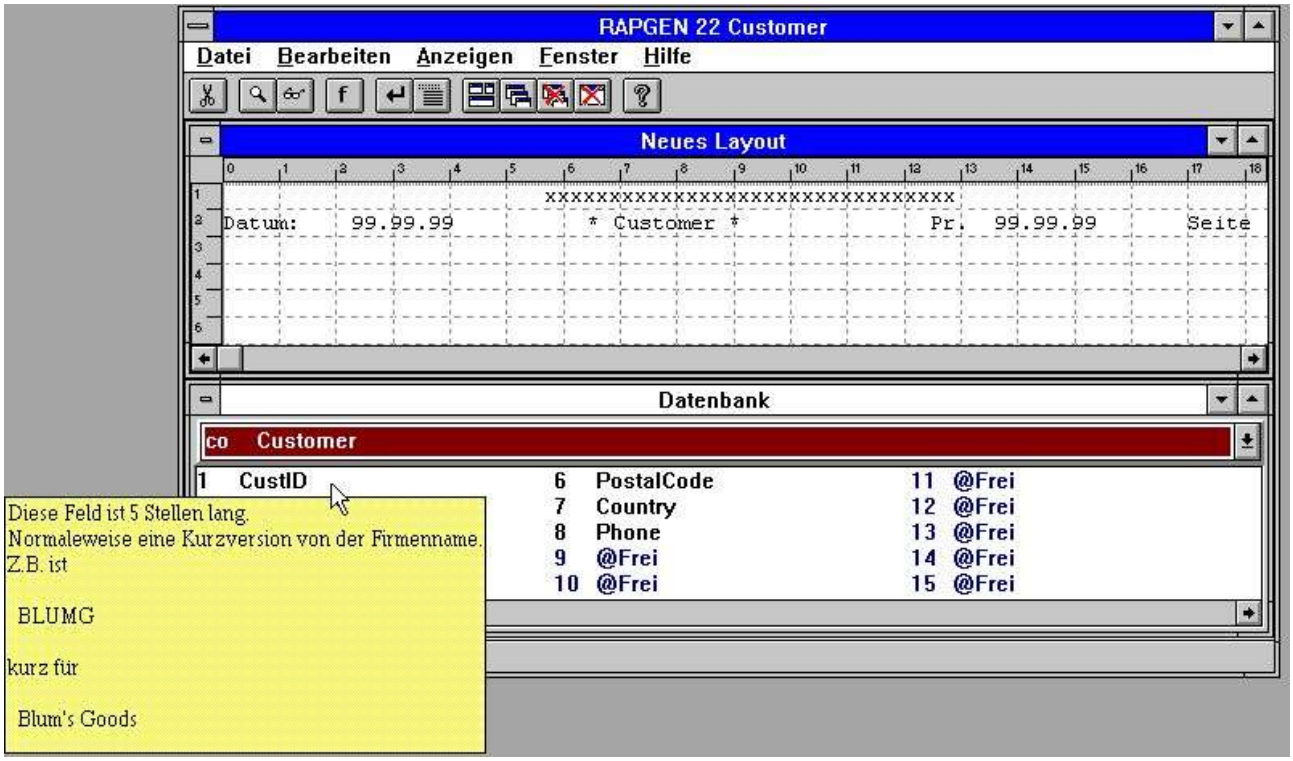
READ(aa.bb) liest die Datei bb unter Verwendung der Verknüpfung aa->bb.
READ(aa.bb.02) liest die Datei bb unter Verwendung der Verknüpfung aa->(bb logischer Index 2)

8.4. Feldbeschreibungen

Kein Data-Dictionary ist vollständig ohne entsprechende Dokumentation. Eine Dokumentation kann für eine Tabelle und für jedes einzelne Feld geschrieben werden. Ist eine Beschreibung für ein einzelnes Feld erstellt, wird diese in den TRIO-Anwendungen verwendet.

Laufende Online-Hilfe in der Feldübersicht.

In den SW-Tools TRIO Anwendungen -Listgenerator, IQ und Datamaster- wird im Datenbankfenster die laufende Online-Hilfe angezeigt, sobald der Cursor über das entsprechende Feld streicht. Beispiel in RAPGEN:



36. Laufende Online Feldbeschreibung

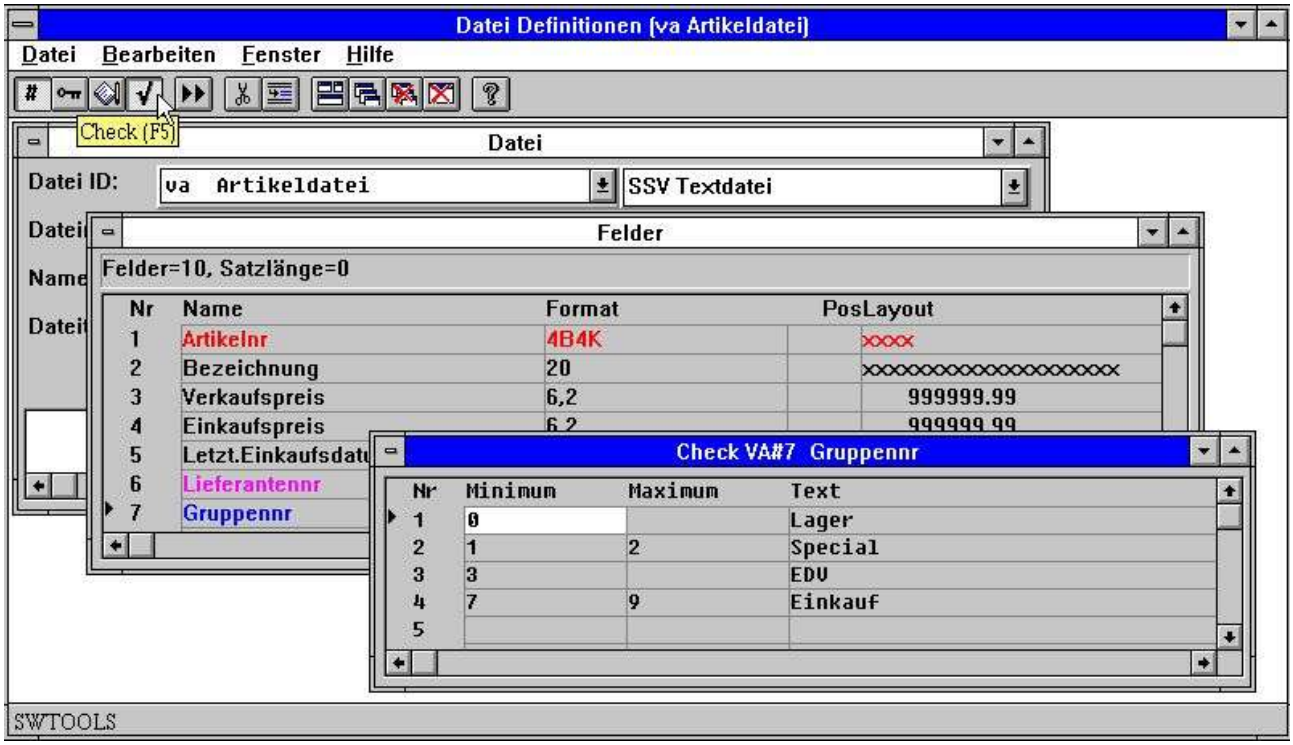
8.4.1. Ausgabe der Feldhilfe

Zum Editieren der Feldhilfsbeschreibung muß das entsprechende Feld markiert werden. Nach Markierung des gewünschten Feldes wählt man HELP im EDIT-Menü.

37. Ausgabe der Feldhilfsbeschreibung

8.5. Feldprüfvorschriften

Prüfvorschriften werden benutzt, um zulässige Feldwerte zu definieren. Dies gilt besonders in SW-Tools DATAMASTER für Eingabeprüfungen. In TRIO Listgenerator und IQ kann diese Funktion zur Verbindung von Texten mit Kodefeldern verwendet werden.
Prüfvorschriften sind definiert als von-bis Werte, verbunden mit einem Text.



38. Ausgabe der Feldprüfvorschriften

8.6. Feld Sondertext

Ein Sondertext verbunden mit einem Feld bildet die zweite Seite des Hilfstextes. Die Eingabe dieses Textes erfolgt wie die Eingabe des normalen Hilfstextes für ein Feld.
Diese Texte sind für zusätzliche Feldparameter und eventuelle Berechnungen reserviert.
Wenn Definitionen von den COMET Dateidefinitionen importiert werden, kann man besondere Texte zusammen mit READ Berechnungen, und Definitionen betr. Installationsparameter erhalten.

8.7. Standard SQL-Namen

Diese Funktion kann benutzt werden, um neue Dateidefinitionen für eine SQL Datenbank einzurichten.
Bei Verwendung von SW Tools TRIO DATAMASTER werden die SQL Namen automatisch gebildet. Werden Tabellendefinitionen von einer ODBC Datenbank importiert, werden die SQL-Namen von den Treibertabellendefinitionen übernommen.

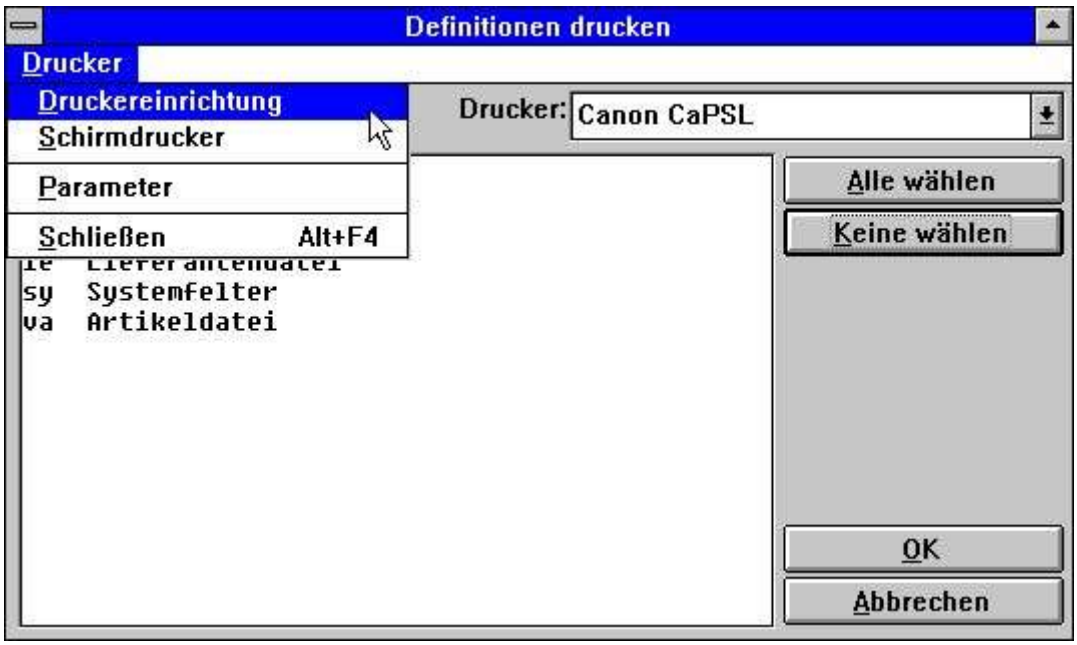
9. Ausdruck der Dokumentation

Um die Dokumentation der Tabellendefinitionen auszudrucken, muß folgende Funktion verwendet werden. Es können eine oder mehrere Tabellen für den Ausdruck markiert werden.

39. Ausdruck der Dokumentation

9.1. Drucker

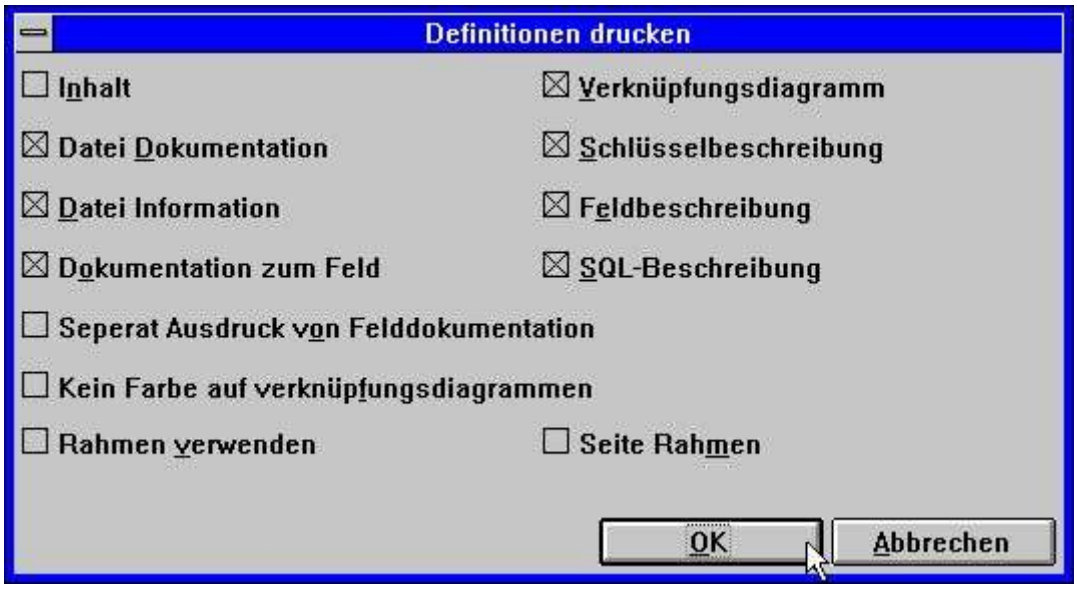
Die Ausgabe kann auch am Bildschirm erfolgen.



40. Wahl des Druckers

9.2. Optionen

Für den Ausdruck der Dokumentation können die Detailebenen mit folgender Funktion gesteuert werden:



41. Optionen für den Ausdruck der Dokumentation

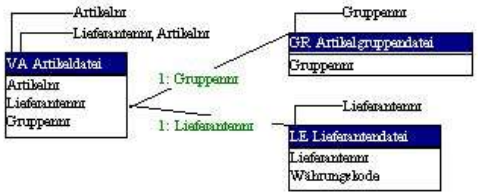
Data-Dictionary (006.006) **SW-Tools** Seite **1**

va - Artikeldatei

Dies ist eine standard Demodatei.

Diese datei steht als SSV-Datei und als BASIC-Datei zu Verfügung.

Name:
Treiber: SSV Textfile
Felder: 10
Schlüssel: 2
Verknüpfungen: 2
Verknüpfungsdiagramm:



Schlüsselname	Datei	Dir.	Schlüssel Definition
Artikelnr		01	1
Lieferantennr		02	6,1
Artikel==>Gruppe	GR	01	7
Artikel==>Lieferant	LE	01	6

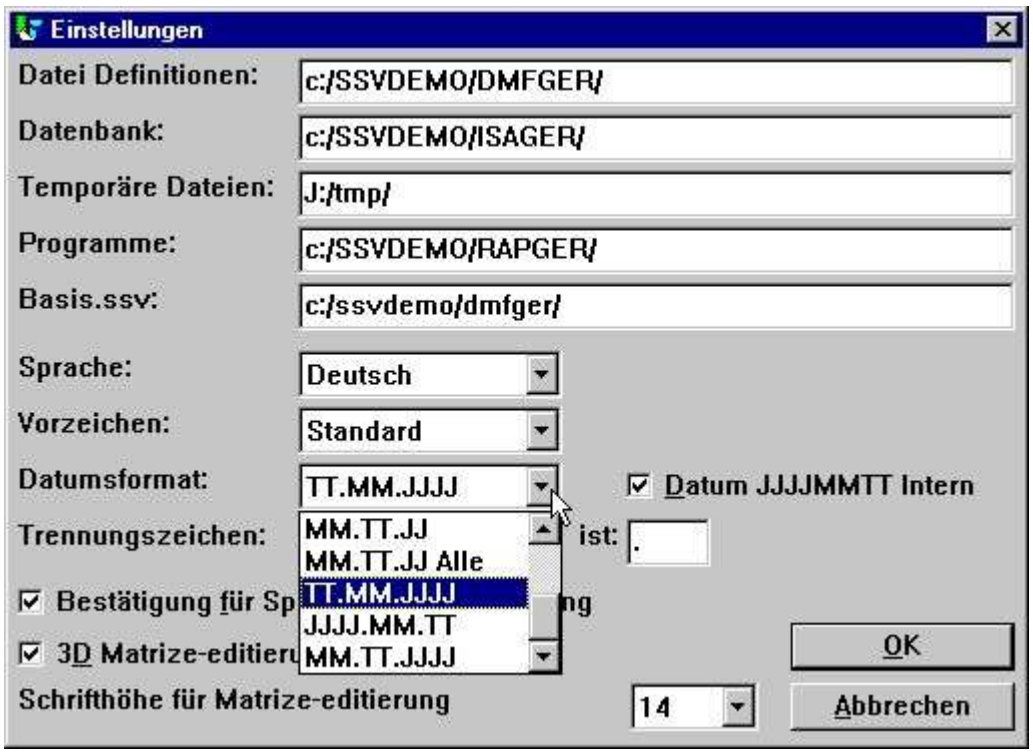
	Feldname	Format	Layout
1	**Artikelnr**	4B4K	xxxx
	Die Artikel Nummer ist der Hauptschlüssel des Datei.		
	Maximal 4 Ziffern kan eingegeben werden.		
2	Bezeichnung	20	xxxxxxxxxxxxxxxxxxx
3	Verkaufspreis	6,2	999999,99
4	Einkaufspreis	6,2	999999,99
5	Letzt.Einkaufsdatum	,6,	99.99.99
6	Lieferantennr	3	xxx
7	Gruppennr	2,	99
	0 = Lager		
	1 - 2 = Special		
	3 = EDV		
	7 - 9 = Nicht auf lager		
8	Bestand	-6,	-999999
9	Alt.Lieferantennr	3,	999
10	Freifeld	-6,2	-999999,99

42. Beispiel einer Tabellendokumentation

ID	Text	Name	Standard ID	Schnittstelle
as	Kundenauftrags Kopf	AF-05000abc	AF-0500000 4 AS	XBasic
au	Kundenauftrags Positionen	AF-05000abc	AF-0500020 4 AF	XBasic
d4	Artikel-Stammdatei 1	LF-06000abc	LF-06000 4 D4	XBasic
d7	Lagerort Datei	LF-06001abc	LF-060010204 D7	XBasic
gr	Artikelgruppendatei			**SSV**
jh	Kunden-Stammdatei	GF-03000abc	GF-03000 4 JH	XBasic
ji	Kunden-Bewegungen	GF-03100abc	GF-03100 4 JI	XBasic

43. Beispiel ein Inhaltverzeichnis

10. Grundeinstellungen



44. Grundeinstellungen

10.1. Directories

Alle SW-Tools Anwendungen benutzen die INIT-Datei RAP.INI. Diese Datei liegt im Startdirectory und beinhaltet die Directories (Adressen) aller Anwendungsdateien.
Das Startdirectory wird in der Windows Ikone festgesetzt. Unterschiedliche Ikonen mit unterschiedlichen Startdirectories können angelegt werden, um z.B. verschiedene Data-Dictionary auf dem gleichen System (PC) zu ermöglichen.
Das RAP.INI Datei enthält folgenden Informationen (Directories):

10.1.1. Dateidefinitionen

In der Dateidefinition wird das Directory für das Data Dictionary angegeben. Beim Definieren einer Tabelle im Data Dictionary generiert das System eine SSV-Datei als

<table id>.SSD

und ändert entsprechend gleichzeitig die Datei

FILES.SSV

Die erste Datei enthält alle Informationen über die einzelnen Definitionen in einer Tabelle. Die zweite Datei wird nur im Zusammenhang mit den Dateiübersichten benutzt.

10.1.2. Datenbank

Diese Adresse betrifft die Standard Datenbankdateien für alle Treiber. Benötigt ein Treiber eine Adresse (path) für seine Dateien, eine solche Adresse ist aber nicht angegeben, wird diese Adresse verwendet. Es handelt sich hierbei um die physische Adresse der Datenbankdateien.

10.1.3. Spool Directories

Soll eine Druckausgabe vorgenommen werden, muß genug Platz für eine zwischenzeitliche Spooldatei vorhanden sein. Die Adresse (path) sollte für ein Directory angegeben werden, in dem ausreichend Platz für eine solche Spooldatei ist.
Wird hier keine Angabe gemacht, setzt das System die Variable TMP bzw. TEMP ein.

10.2. Sprache

Deutsch, Englisch und Dänisch wird z.Zt. vom System unterstützt. Normalerweise enthält die Installationsdiskette jedoch nur eine einzelne Sprache. Bitte kontakten Sie SW-Tools.

10.3. Vorzeichen

Die Option für das Vorzeichen betr. die Ausgabe numerischer Werte. Es stehen drei Optionen zur Verfügung:

- Standard
- vorangestellt
- nachgestellt

Die Standardoption erlaubt die Festlegung des Vorzeichens in dem Feldformat, also

-9,2 oder
9,2-

Die zwei anderen Optionen bestimmen fest die Position des Vorzeichens (unabhängig von dem jeweiligen Feldformat).

10.4. Datum Ausgabeformat

Datumfelder können als Tag/Monat/Jahr oder Jahr/Monat/Tag formatiert werden. Die Standardoption entspricht dem Format Tag/Monat/Jahr.
Die Möglichkeit der Datumsausgabe TT.MM.JJJJ und MM.TT.JJJJ (amerikanisch) wurde hinzugefügt.

```
0 = Keine Konvertierung
1 = ,6,          -> TT.MM.JJ
2 = ,6,          -> JJ.MM.TT
3 = ,6, und ,8, -> TT.MM.JJ,  ,8,& -> TT.MM.JJJJ
4 = ,6, und ,8, -> JJ.MM.TT   ,8,& -> JJJJ.MM.TT
5 = ,6,          -> MM.DD.JJ
6 = ,6, und ,8, -> MM.DD.JJ   ,8,& -> MM.TT.JJJJ
7 = ,6,          -> TT.MM.JJ,  ,8,  -> TT.MM.JJJJ
8 = ,6,          -> JJ.MM.TT   ,8,  -> JJJJ.MM.TT
9 = ,6,          -> MM.TT.JJ   ,8,  -> MM.TT.JJJJ
```

Beachten Sie Bitte auch die Option TT.MM.JJ-All mit der Möglichkeit, auch die Ausgabe ,8, JJJJMMDD zu überschreiben.
Ein ,6, Feld wird für einen besseren Ausdruck als ,6,& behandelt, also Unterdrückung vorangestellter Nullen.

10.5. Trennzeichen

Trennzeichen für numerische Felder können festgelegt werden für:

- Dezimalpunkt
- Tausender Trennung
- Datum Ausgabe

Die Änderung des Trennzeichen bewirkt folgenden Wechsel:

123,456,789.12

in

123.456.789,12

10.6. Änderungsbestätigungen

Ist dieser Punkt markiert, muß Änderungen in Dateidefinitionen, Listen und IQ-Programmen gesondert bestätigt werden.
Ist dieser Punkt nicht markiert, werden die Änderungen ohne weitere Mitteilung gespeichert.

10.7. Format für interne Datumberechnung

- BITTE VORSICHT, WENN SIE DIESE OPTION AKTIVIEREN -

Wird das Datum in der Kurzform JJMMTT gespeichert, müssen Sie dies bei der Berechnung in Listen und IQ Programmen mit einem Datum berücksichtigen.
Bisher kann eine Berechnung mit auf dem ,6, Feld #47 wie folgt geschrieben sein:

IF 19000000+#47>#DD LET #99=1

Aktivieren Sie die JJJJMMTT Berechnungsoption, wird jedes Kurzdatum beim Lesen(Schreiben) von Dateien in das lange Format JJJJMMTT Jahr 19xx/20xx umgewandelt.
In diesem Fall müssen Sie also Ihre Berechnung wie folgt ändern:

IF #47>#DD LET #99=1

Es ist natürlich vorzuziehen, daß die Umwandlung automatisch erfolgt. Bitte seien Sie sehr sorgfältig, wenn Sie diese Option benutzen.

10.7.1. Jahr 2000

Da TRIO mit einer Reihe unterschiedlichen Dateisystemen zusammenarbeiten kann, treten alle möglichen Formen von Jahr 2000 Verarbeitungen auf.
Im Zusammenhang mit der X-Basic COMET Unterstützung wurde das Datum in reiner alphanumerischen Form und gepackt in Indexfeldern hinzugefügt (siehe Packungstypen).

10.7.1.1. Sortieren nach Datum

Sortiert man nach einem Kurzdatum ,6, Feld wird die Sortierdatei mit ,8, Feldern eingerichtet. Die gleiche Regel gilt für Gruppensummen im Listgenerator.
Jahr 2000 wird auch korrekt sortiert, wenn ORDER BY in einem ,6, Kurzdatumfeld benutzt wird.

10.7.1.2. DOS Datum

DOS und gleichartige Systeme geben den Wert 100 für das Jahr 2000.
Ein Datumfeld ,6, oder ,8, mit einem Wert zwischen 1.000.000 und 1.999.999 wird als Jahr 2000 angenommen.

10.7.1.3. Jahr 2001 Problem

Beachten Sie bitte, daß die FNY Funktion Probleme bekommt, wenn ein TTMMJJ Datum übergeben wird:

FNY(31.03.98) = 1998.03.31 ok
FNY(31.03.00) = 2000.03.31 ok
FNY(31.03.01) = 2031.03.01 JJMMTT ist vorzuziehen anstelle von TTMMJJ

10.7.1.4. Jahr 2005 Problem

In gepackten BASIC Schlüsselfeldern wird das erste gepackte Zeichen als Jahr 2000 angenommen, wenn dies keine Ziffer ist. Fünf solche Zeichen stehen zur Verfügung: Leerstelle, Plus, Komma, Minus, Punkt, also für Jahr 2000 bis 2004.

10.7.1.5. Jahr 2010 Problem

In ungepackten BASIC Schlüsselfeldern wird das erste Zeichen als Jahr 2000, wenn dieses ein Fragezeichen (?) ist. Dies ermöglicht die Jahre 2000-2009.
TRIO behandelt ?@ABCDEF... auch als Jahr 200x,201x,202x,... .

10.7.1.6. Jahr 2100 Problem

Nach Addieren des Jahres zum Kurzdatum JJMMTT wird Jahr 2000 angenommen, wenn JJ kleiner als 50 ist.
Überschreitet das Systemdatum das Jahr 2050, gilt die gleiche Regel für das nächste Jahrhundert.

10.7.1.7. Datum Eingabeprüfung

Bei Eingabe eines Datums gelten die folgenden Regeln in der gegebenen Reihenfolge:

```
   0. Eingabe 0        -> 0
   1. Eingabe TT       -> TT+MMJJJJ von Systemdatum
   2. Eingabe TTMM     -> TTMM+JJJJ von Systemdatum
   3. Eingabe TTMMJJ   -> TTMMJJ+TT von Systemdatum
   4. Eingabe TTMMJJJJ -> TTMMJJJJ, wird auch in ,6, Feldern akzeptiert
   5. Datumprüfung entspr. Grundeinstellung
   6. Falls unkorrekt, Prüfung nach: TTMMJJJ, JJJJMMTT und MMTTJJJJ
.
```

10.7.1.7.1. IQ Transaktionswahl, Datumeingabe

Benutzen Sie die Transaktionswahl in IQ für die Datumseingabe, wird die Eingabe und das Datumfeld an die FNY Funktion weitergereicht, um sicherzustellen, daß das Jahr richtig gesetzt ist.

10.7.1.7.2. CHAIN Funktion, maskierte Datumeingabe

Benutzen Sie CHAIN, um eine Liste zu starten, wird das Datum, das von der Funktion weitergereicht wird, wie ein bei Start manuell eingegebenes Datum behandelt.

10.8. FDF Startparameter

Das FDF Modul kann aus Windows mit folgenden Parametern gewählt werden:

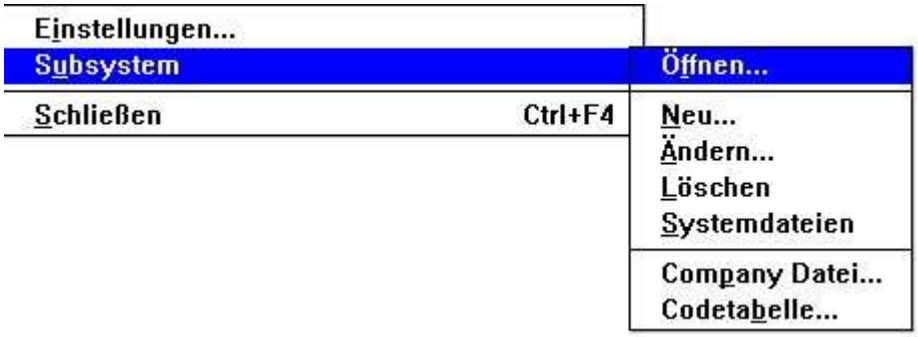
```
RAPFDF -d       Dateien werden als DATAMASTER Dateien definiert
RAPFDF -d7      DATAMASTER Standarddateityp ist 7
```

11. Subsysteme

Mit Hilfe des Preferencemenü wird ein System durch Setzen der Parameter in der RAP.INI Datei im aktuellen Arbeitsdirectory definiert. Diese Datei wird bei Programmstart gelesen.
Ein Subsystem kann während eines Programmablaufes aufgerufen werden. Die Einstellungen des Untersystems ersetzen bzw. modifizieren die aktuellen Systemparameter.
Typisch hierfür kann das Gruppieren von List- oder IQ-Programmen sein. Dies kann durch Duplizieren der Listgeneratorikonen in Windows in ein anderes Arbeitsdirectory (eine neue RAP.INI Datei) oder durch Anlage eines Subsystems in einem anderen Directory eingefügen.

11.1. Subsystem Menü

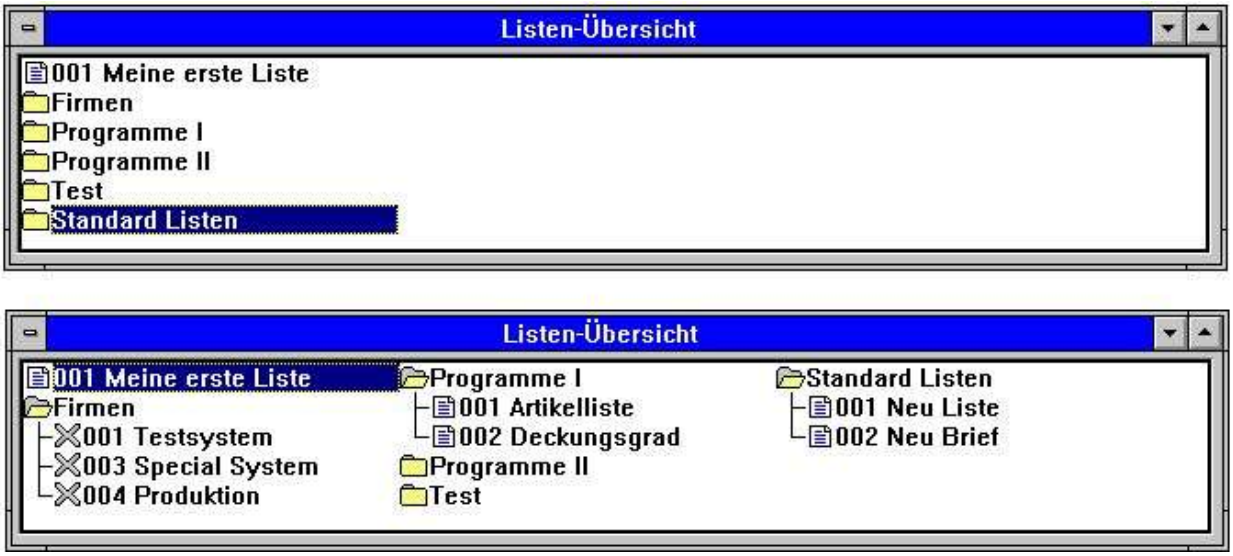
Subsysteme werden gewählt, angelegt/geändert und gelöscht über das zugehörige Menü.



45. Subsystem Menü

11.2. Öffnen eines Subsystems

Diese Funktion wählt das gewünschte Subsystem, wie die Programmwahl in RAPGEN oder IQ.



46. Öffnen eines Subsystems

Subsysteme werden geöffnet/geschlossen durch Klicken auf die entsprechende Ikone oder Namen. Jedes Subsystem kann ein oder mehrere Eingänge besitzen. Ein Eingang kann ein Programm oder eine Datenvorgabe für das spätere Programm, wie z.B. eine Firmennummer, sein.

11.3. Anlegen eines Subsystems

Die Funktion Neu, Änderungen und Löschen betreffen entweder ein Subsystem oder einzelne Punkte innerhalb dieses Systems. Dies ist von der gewählten Funktion ab. Ein einzelnes Programm kann ebenso modifiziert werden

47. Anlegen eines Subsystems

Normalerweise sind für die Anlage nur wenige Informationen notwendig. Informationen, die nicht gesondert angegeben werden, werden von den Standardvorgaben bzw. von einem anderen Subsystem übernommen.

11.3.1. Name

Der hier eingegebene Name (Freitext) wird zusammen mit der entsprechenden Ikone des Subsystems angezeigt.

11.3.2. Paßwort

Wird hier ein Paßwort eingegeben, muß dies jeweils bei Programmstart angegeben werden.

11.3.3. Firmanummer

Eine Firmanummer kann als 001 oder 444 eingegeben werden. Hierüber werden die Dateinamen bei Dateieröffnung kontrolliert, siehe weiter unten.

11.3.4. Adresse der Dateidefinitionen

Jedes Subsystem kann seine eigenen Dateidefinitionen beinhalten. Hierfür muß hier die Adresse dieser Definitionen angegeben werden. Erfolgt hier keine Eingabe, werden die Standarddefinitionen benutzt.

11.3.5. Standardadresse der Datenbank

Dieses Feld überschreibt die Adresse der Datenbank, die in den Standardvorgaben angegeben ist. Die Adresse wird normalerweise nur benutzt, wenn die Datenbank auf einem PC installiert ist.

11.3.6. Directory der Listen

Ein Subsystem kann zur Aufteilung unterschiedlicher Listen in logische Einheiten genutzt werden. In diesem Falle muß jede Liste ein eigenes Directory besitzen.

11.3.7. Datenbankvorgabe

Dieses Feld bestimmt den Datenbanktyp und sollte normalerweise frei gelassen werden.
Die Dateien können als X-Basic Dateien definiert sein. X-Basic sei Dateityp 2 (zweite Zeile) in der BASIS.SSV Datenbank Interfacedatei und sei verbunden mit dem Server 200.09.0.1.
Wird eine Datenbankvorgabe

2=3

gemacht, wird Dateityp 3 anstelle von Typ 2 für alle Dateien benutzt. Diese können dann einen anderen Server (200.0.0.2) oder andere Dateisysteme, wie z.B. XNet, benutzen.
Mehrere Vorgaben können wie

2,7-8=3,9=4

angegeben werden. Sollen alle Typen in einen bestimmten Typ geändert werden, kann

***=3**

eingegeben werden.

11.3.8. Beschreibung

Hier kann ein freier Text eingegeben werden. Dieser Text wird angezeigt, wenn man mit die Maus über den Namen des Subsystems in der Programmwahl führt.

11.3.9. Module

Hier kann angegeben werden, ob das Subsystem nur aus RAPGEN, aus IQ oder aus beiden Systemen anwählbar sein soll.

11.3.10. Typ

Es gibt grundsätzlich zwei Typen von Subsystemen. Zum einen das normale Subsystem, zum anderen das sog. übergreifende Subsystem. Verläßt man ein normales Subsystem, sind alle Informationen dieses Systems wieder vergessen. Bei einem übergreifenden System jedoch bleiben eine Reihe der Informationen für die nachfolgenden Programme in Erinnerung.
Ein normales Subsystem kann als Programmauswahl betrachtet werden. Wird ein neues Subsystem aufgerufen, erhält man eine andere Programmauswahl in einem neuen Directory.
Ein übergreifendes Subsystem dagegen kann als Parameterwahl für alle folgenden Programme gesehen werden. Arbeitet man z.B. mit mehreren Firmen, können diese als übergreifende Subsysteme eingerichtet werden. Wählt man jetzt eine Firma aus, werden die im Subsystem gemachten Vorgaben für alle Programme verwendet, bis eine neue Firma angegeben wird.
Ein übergreifendes Subsystem kann in verschiedene Kategorien, abhängig von der Verwendung, eingeteilt werden.

Nr	Name	Verwendung	Datei
0	Normal	Nicht übergreifend	-
1	Firma	Modifiziert die Firmanummer	COMPANY.SSV
2	Datadict	Wahl eines anderen Data-Dictionary (Dateidefinitionen)	DATADICT.SSV
3	Listen	Wahl eines anderen Listdirectorys	REPORTS.SSV
4	Datasyst	Wahl eines anderen Dateityps für Dateien in diesem Data Dict.	DATASYST.SSV
5	Basis	Wahl eines Dateityps von BASIS.SSV und Vorgabe alle Dateien	BASIS.SSV
6	Dmsystem	Wahl eines Subsystems aus der Subsystemdatei	DMSYSTEM.SSV
7	Standard	Ermöglicht Änderungen in Standard Neue Liste und Neuer Brief	-

11.3.10.1. Subsystemtype 'Firma'

Wird in einem übergreifenden Subsystem ein Punkt angewählt, bewirkt dies keinen Programmstart sondern es wird eine Mitteilung ausgegeben, die angibt, welcher Punkt aktiviert ist:

48. Firmenwahl

Die Parameter für dieses Subsystem werden aktiviert, wenn eine Liste/Programm oder ein anderes Subsystem gewählt wird, es sei denn, daß eine solche Wahl die Parameter neu definieren. Im Beispiel wird die Firma 004 aktiviert. In allen folgenden Subsystemen sollte die Angabe der Firma freigelassen werden, um diese nicht zu überschreiben.

11.3.10.2. Subsystemtype 'Datadict, Listen und Datasyst'

Jedes Subsystem kann auf sein eigenes Data-Dictionary, seine eigenen Listen oder sein eigenes Datenbankinterface verweisen. Es kann auch ein übergreifendes Subsystem wie für die Firmen definiert werden.

11.3.10.3. Subsystemtype 'Basis'

Dieser Typ bestimmt übergreifend ein Datenbankinterface aus des BASIS-Datei. Die Informationen in der BASIS-Datei sollten nur über das Datenbankinterface-Menü geändert werden (nicht über 'ändern des Subsystems').

11.3.10.4. Subsystemtype 'Dmsystem'

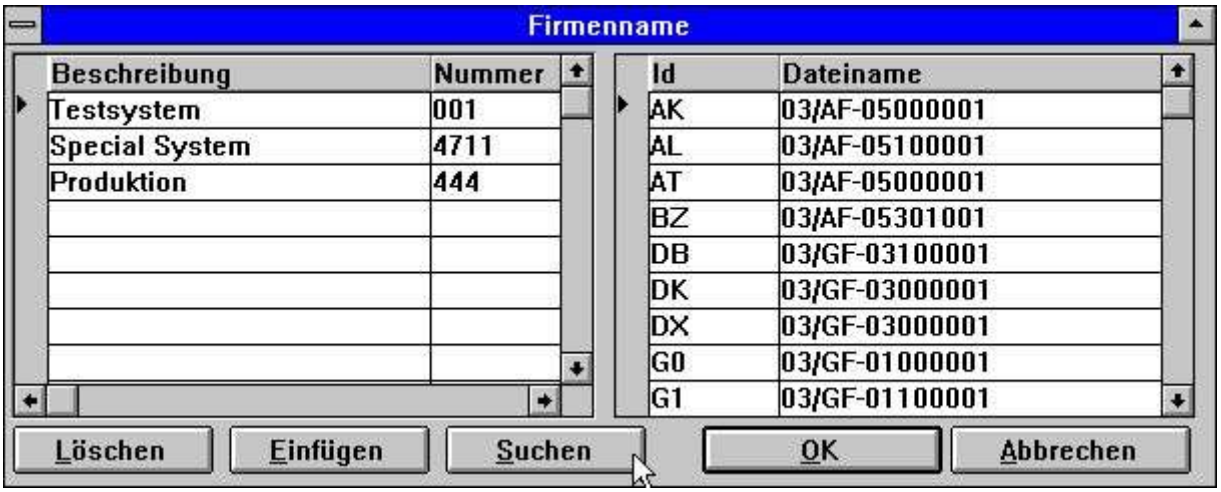
Ein normales Subsystem ist nicht übergreifend. Die Informationen werden bei Verlassen des Subsystems 'vergessen'. Mit dieser Funktion kann ein normales Subsystem gewählt werden, wobei jedoch die Parameter übergreifend werden.

11.3.10.5. Subsystemtype 'Standard'

Hier werden die Standardüberschriften für 'Neue Liste' und 'Neuer Brief' gehalten. Hier können z.B. Firmenname oder andere individuelle Informationen abgelegt werden.

11.4. Dateinamen für Firmen

Hier können auf der linken Seite Firmenname und Nummer definiert werden. Dies kann auch über das Subsystemmenü erfolgen.



49. Dateinamen für Firmen

Bei Klick auf ein bestimmtes Subsystem wird auf der rechten Seite eine Liste der zugehörigen Dateinamen angezeigt. Es können jetzt Dateinamen für die firmenspezifischen Dateien eingegeben werden.
Die Dateinamen werden in <company>.KOM, z.B. 001.KOM, gespeichert. Es können jetzt unterschiedliche Dateien für jede Firma benutzt werden.

11.4.1. Suchen nach Firmendateinamen

Bei der Nutzung verschiedener Firmen-Dateien , können alle LU's durchsucht werden. Hierzu müssen Sie den Button SEARCH aktivieren.
Alle BASIC-Dateien, die mit einer 2-stelligen LU + Dateiname, z.B. 90/GF-03000abc, definert sind, werden danach geprüft , ob eine LU 00-99 in der gewählten Firma vorhanden ist. Die erste gefundene LU wird zugeordnet.
Beachten Sie bitte, daß hierfür als Minumum die Version (006.003) für den Server erforderlich ist.

11.4.2. DOS Dateinamen für BASIC-Dateien

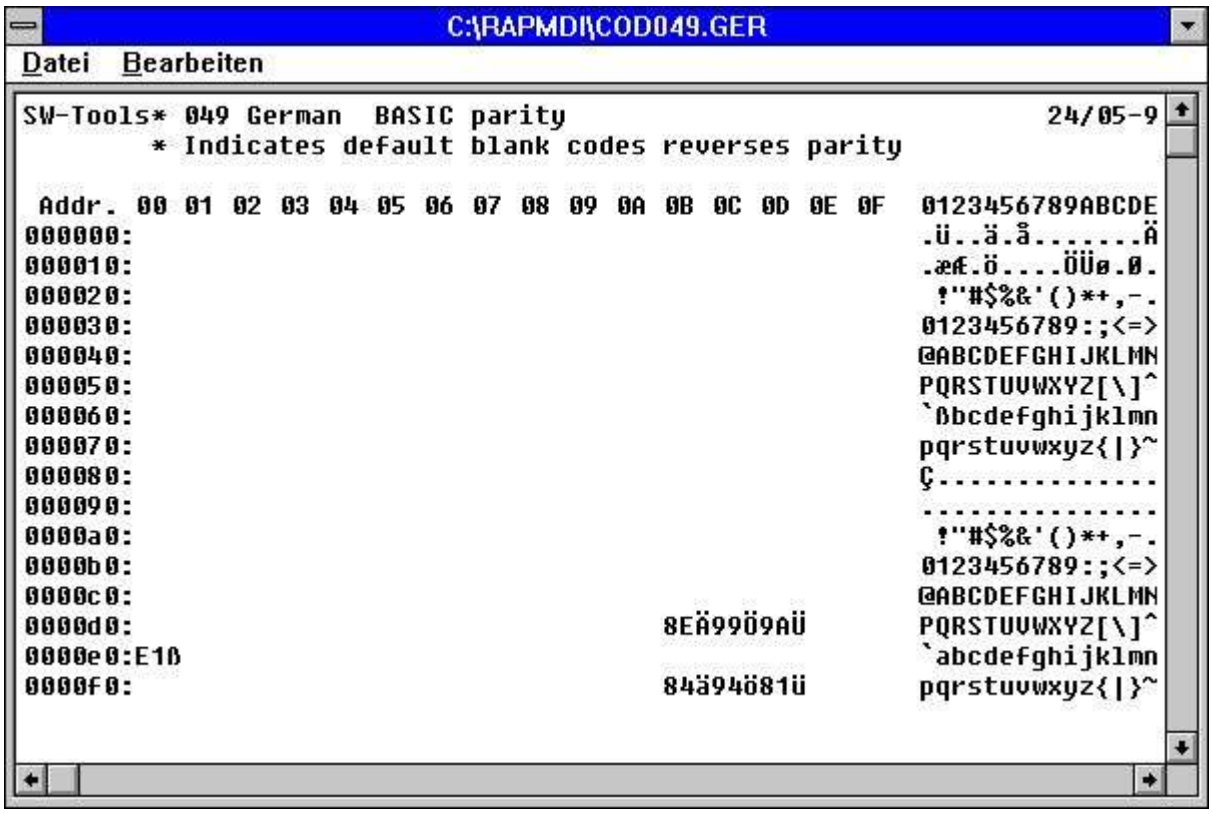
Arbeitet man Offline vom Server mit einem X-Basic System, können Probleme mit der Länge des Dateinamens entstehen. Es wurde deshalb jetzt folgendes festgelegt: Wird eine Basic-Datei auf einem DOS System nicht gefunden, wird der Dateiname wie folgt behandelt:

1. Alle Punkte werden entfernt.
2. Besteht der Name aus mehr als 8 Zeichen, wird nach dem achten Zeichen ein Punkt gesetzt.
3. Der Name wird auf das Format 8.3 Zeichen reduziert.

Dieses gibt die Möglichkeit, eine Serverdatei GF-03000001 als lokale Datei GF-03000.001 zu behandeln.

11.5. Kodetabellen

Eine Kodetabelle wird in der Datei CODnnn.LLL, z.B. COD013.ENG, gespeichert, wobei nnn die Nummer der Tabelle und LLL die entsprechende Sprache angeben.
Die Kodetabellen 000-009 sind vom System belegt.
Kodetabellen können binär oder in ascii-Textdateien gespeichert werden. Im letzten Falle uns die Markierung 'SW-Tools' in der ersten Zeile eingetragen werden. Die Datei kann mit dieser Funktion editiert werden. Soll ein bestimmtes Zeichen nicht konvertiert werden, kann dieses in der Kodetabelle als Leerstelle auftreten. Dies gibt Ihnen einen verbesserten Überblick.



50. Kodetabelle zum Lesen von BASIC-Dateien

Die Kodetabelle besteht aus einer Adresse, gefolgt von den hexadezimalen Kodes. Alle anderen Zeilen sind Kommentare.
Die Kodetabelle wird zum Übersetzen beim Lesen einer Datei von der Festplatte verwendet.
Ab Version (006.xxx) ist die Paritätkennung in den Kodetabellen enthalten. Dies bedeutet, dass die Tabellen 1 und 2 (dänisch und deutsch) automatisch die Tabellen 045 und 049 verwenden.
Um diese Möglichkeit auch in BASIC Kodetabellen zu nutzen, müssen Sie ein * (Asterix) nach SW-Tools in der ersten Zeile angeben. Es werden jetzt alle Paritäten gewechselt. Das korrekte Zeichen kann als Bemerkung nach dem hexadezimalen Wert angegeben werden, oder der hexadezimale Wert kann mit dem Zeichen selbst erstattet werden.

11.5.1. Nummern der Kodetabellen

Die Namen der Kodetabellen entsprechen den Länderkodes für das Telefonieren.

001 dänisch UNIX eingeschl.
002 deutsch UNIX eingeschl.
003 ODBC OemToAnsi eingeschl.
004 Parity plain 1:1 eingeschl.
010 ISO8850
011 Dataflex
012 EBCDIC
013 ODBC OemToAnsi
044 englisch Basic parity
045 dänisch Basic parity
046 schwedisch Basic parity
049 deutsch Basic parity
099 Sortieren vielsprachig

11.5.2. Kodetabelle für Sortieren

Beim Sortieren in RAPGEN oder SW-Tools ODBC Treiber wird die Kodetabelle für Sortieren benutzt. Hierdurch werden länderspezifische Zeichen (z.B. ä, ü, ö) korrekt sortiert. Dies gilt auch für Zeichen mit Akzenten.

11.6. Systemdateien

Um dieses System benutzten zu können, müssen die Definitionen der Systemdateien installiert sein. Hier werden die Systemdateien für das Data-Dictionary, Subsysteme, Listen und IQ-Programme definiert.

12. Datenbank Treiber

Das Data-Dictionary unterstützt eine Reihe von Datenbank Treibern. Weitere sind in Vorbereitung.
Einige dieser Treiber sind von SW-Tools entwickelt und gewährleisten generell schnelleren Zugriff auf Dateisysteme. Es ist jedoch zu beachten, daß hierbei einige Restriktionen betr. Index-updates in Kauf genommen werden müssen.
Werden andere Zugriffsroutinen verwendet, müssen diese separat erworben und installiert werden.
ANMERKUNG: SW-Tools Treiber unterstützen vorwärts und rückwärts Suchen in Indextabellen. Andere Treiber unterstützen nur die eingebauten Funktionen, z.B. kann IQ bei Verwendung des CTRAS Treibers keine Übersichten in rückläufiger Ordnung ausgeben.
Auch wenn das Ajourführen von Dateien möglich ist, geschieht dies auf eigenes Risiko des Anwenders. SW Tools kann keine Verantwortung für Fehlverhalten übernehmen, wenn solche Programme eingesetzt werden.

12.1. SSV Treiber für Textdateien

Der SSV Treiber für Textdateien bestimmt das interne Dateiformat für alle SW-Tools Produkte. Dieser Treibern sollte deshalb immer als erster Treiber vorhanden sein.
In einer SSV Textdatei ist jeder Satz durch ein Carriage Return, und jedes Feld durch ein Semikolon getrennt. Die Dateien können mit einem beliebigen Texteditor ausgegeben werden.
SSV Dateien sind extrem schnell bei relativ kleinen Dateien. Die Dateien können sowohl auf dem lokalen PC als auch auf einem Server installiert sein. Es kann jede beliebige Kodetabelle verwendet werden. Rückschreiben von Daten ist implementiert und zugelassen, sollte jedoch nur von einem Anwender zur Zeit vorgenommen werden.
Handelt sich um umfangreiche Dateien bzw. soll ein Rückschreiben in einer Multiuser-Umgebung erfolgen, sollte man ein echtes Datenbanksystem wählen.

12.1.1. Tabellenname

Der physische Dateiname wird aus dem Standardpath und dem Tabellennamen für den Treiber generiert. Die Dateiextension ist normalerweise .SSV, falls anders nicht angegeben wird. Wird kein gesonderter Dateiname angegeben, wird hierfür die Datei-ID benutzt, z.B. greift dann die Tabelle va auf die Datei va.ssv zu.

12.1.2. Feldformat

Alle Felder werden als Textfelder gespeichert, können aber als numerische Felder definiert werden. Die Option für das Packen wird nicht benutzt. Bytenummern sind hier bedeutungslos.

12.1.3. Indexbeschreibung

Die einzelnen Sätze sind physisch immer entsprechend der ersten Indexdefinition sortiert. Wird ein sekundärer Index verwendet, erfolgt die Sortierung hiernach nur bei Bedarf.

12.2. ODBC Treiber

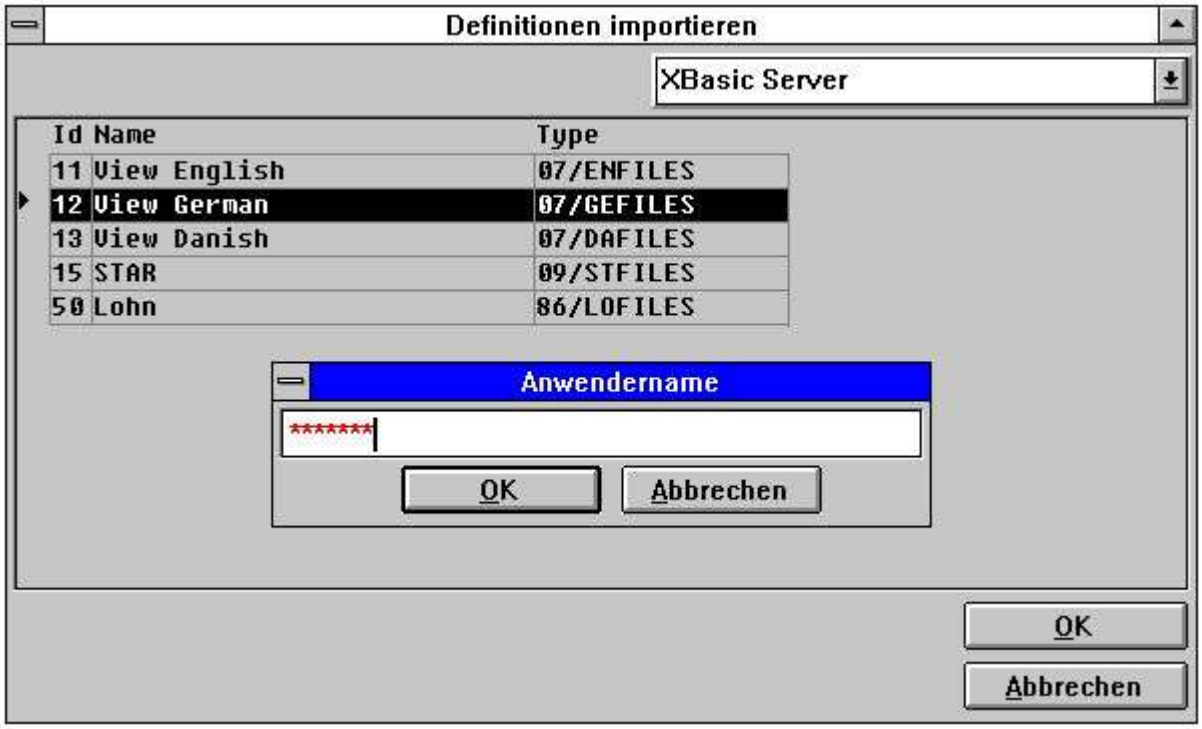
Der ODBC Treiber, installiert unter Windows, ist für das Data-Dictionary verfügbar. Ein Installationsbeispiel finden Sie im Abschnitt 'Treiber Schnittstelle'.

12.2.1. Kodetabelle

Die Kodetabelle solle normalerweise ODBC (Oem auf Ansi) entsprechen, kann jedoch abhängig von der Anwendung, die die Tabellen definiert hat, abweichen.

12.3. BASIC Treiber, generell

Der BASIC Treiber entspricht den Spezifikationen in den VIEW Handbüchern. Man kann Dateidefinitionen direkt aus VIEW importieren (PUT, gefolgt von 'Get Standard Definitionen). Auch COMET Textdateien können direkt geladen werden.
Über die Funktion 'Import ODBC Definitionen kann ein komplettes VIEW System, einschließlich Firmanabhängiger Dateinamen, vom Server installiert werden. Diese Funktion setzt ein Paßwort (BASIC) voraus, wenn ein gesamtes System, also nicht nur Teilsysteme, importiert werden soll.



51. 'Import ODBC Definitionen für einen BASIC Treiber

12.3.1. Feldformate

Das BASIC Dateisystem speichert Feldwerte als BASIC-Variable, wobei die folgenden Feldtypen verwendet werden können:

T1 1% = 1-Wort, max. 7999, keine Dezimalstellen
T2 2% = 2-Wort, max. 6 Stellen
T3 3% = 3-Wort, max. 10 Stellen
T4 4% = 4-Wort, max. 14 Stellen
T5 5% = 5-Wort, max 18 Stellen, nur UNIBASIC

Der Variablentyp wird anhand des logischen Formats automatisch berechnet. Z.B. das Feldformat 2, wird zu einer 1% Integer, das Format 9,2 zu einer 3% Integer usw.

12.3.2. Gepackte Felder

Feldwerte in einem BASIC Dateisystem können gepackt werden. Ein Feldformat kann deshalb einen Packungstyp verlangen. Folgende Packungstypen werden unterstützt:

P Wert gepackt mit CALL 60
P1 wie P
P2 wie P, keine Dezimalstellen
P3 alphanumerisch, mit Dezimalstellen
P4 alphanumerisch, keine Dezimalstellen
P5 wie 1%, keine Dezimalstellen

12.3.3. Indexbeschreibungen

Wie oben beschrieben werden Indizes als Text gespeichert, und können Bestandteil des Datensatzes sein, oder auch nicht. Konstanten als Teil eines Schlüssels und Packen von Schlüsseln werden unterstützt.

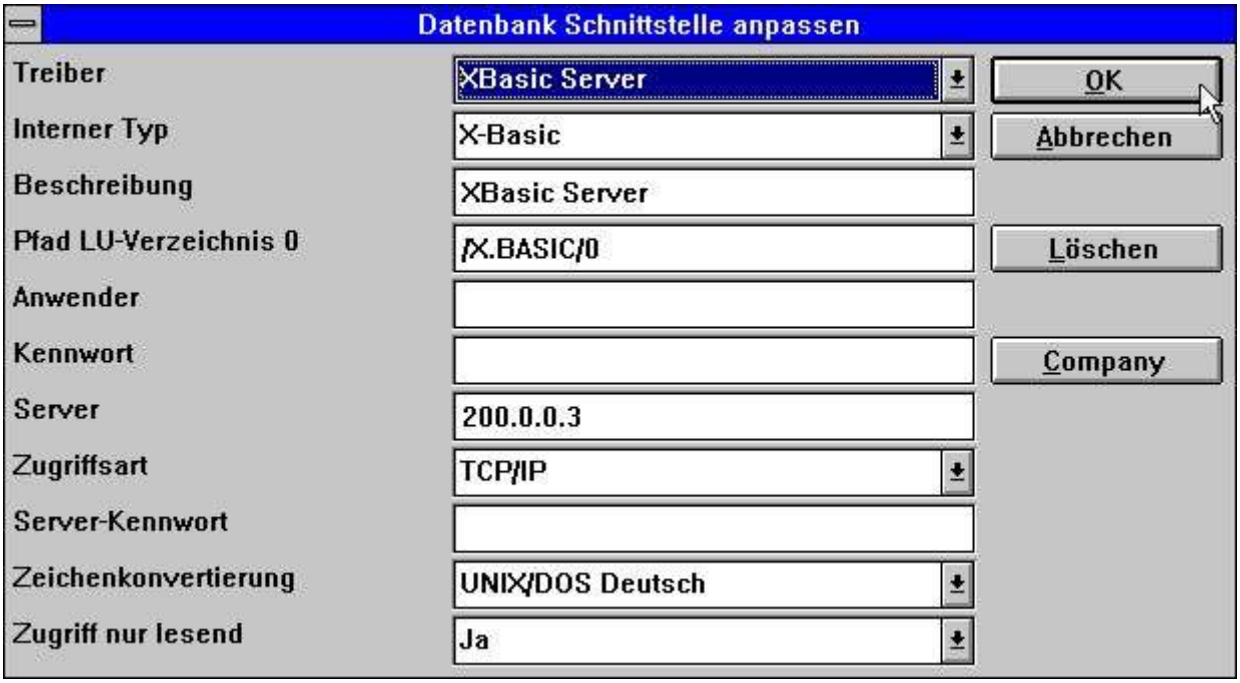
12.4. BASIC Schnittstellen

Im folgenden werden die zur Verfügung stehenden BASIC Schnittstellen beschrieben:

Typ	TRIO Schnittstelle	Andere Software vorausgesetzt	WRITE	Bits
XBasic-Unix	XBasic			
	CTRAS (Library)		Write	
	CTRAS	CTRASX.DLL	Write	nur 16
XBasic-NT	X-Basic			
Quattro	CTRAS	CTRASQ.DLL	Write	nur 16
	Quattro (FTP)	SSQ		32 empfohlen
X/Net	X/Net			
Netbasic	Netbasic			
	CTRAS	CTRASQ.DLL (Netbasic)	Write	nur 16
Unibasic	Unibasic			
Surfbasic	Unibasic			
CX-Basic	Unibasic			
OpenBasic	OpenBasic			

12.5. X/Basic Treiber

Der X/BASIC Treiber kann für den Zugriff auf X-Basic Dateisystem auf einem UNIX-System oder einem PC benutzt werden.
Rückschreiben in Dateien ist beschränkt auf Nicht-Indexfelder. Datensätze können weder eingefügt noch gelöscht werden.

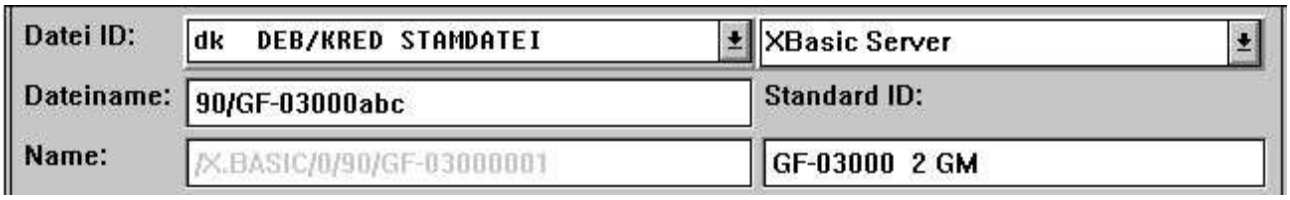


52. Einstellen eines X/BASIC Treibers

Normalerweise befindet sich die Datenbank auf einem UNIX-System. Der Treiber muß dann entsprechend Windows Socket TCP/IP eingestellt werden.

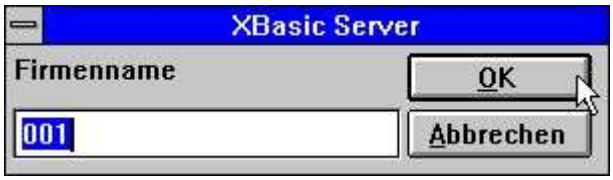
12.5.1. Dateiname

Der physische Dateiname wird entsprechend dem Standard LU0-Path für den Treiber und den Tabellennamen generiert.



53. X.BASIC Dateiname

12.5.2. Firmennummer

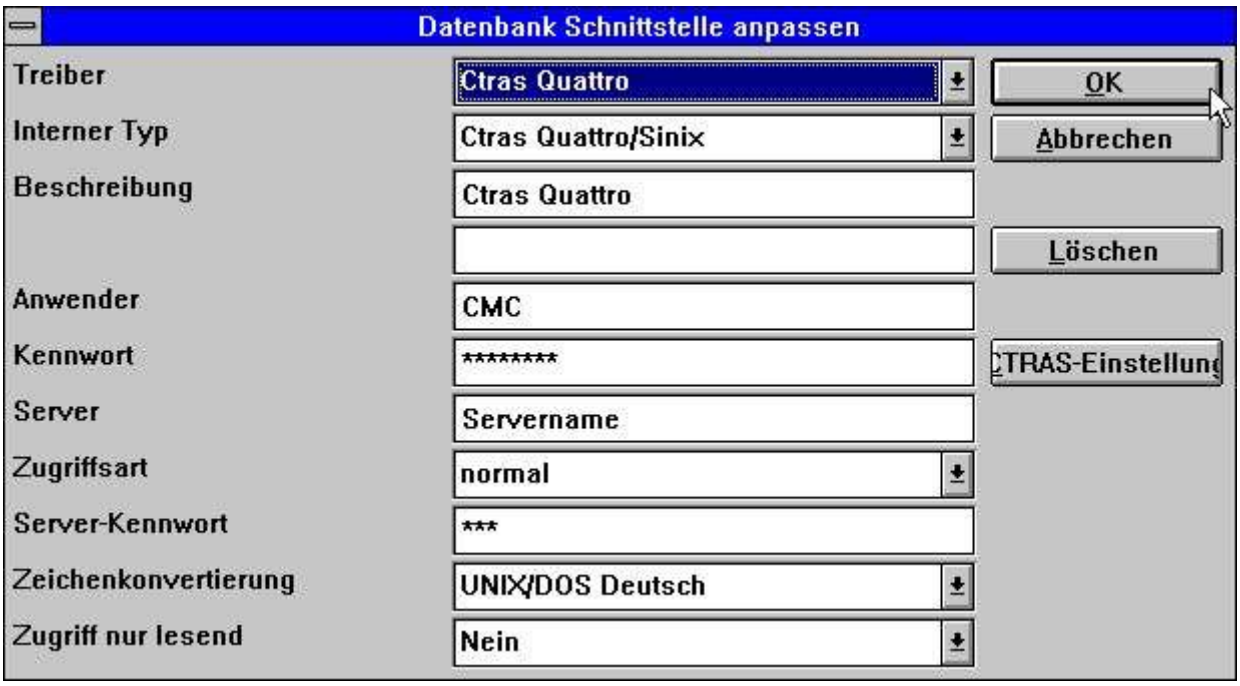


54. Standard Firmennummer

Die Firmennummer entspricht normalerweise einem 3-stelligen Wert zwischen 000 und 999. Dieser Wert wird verwendet, wenn der Tabellenname den Verweis 'abc', wie z.B. 90/GF-03000abc, enthält.

12.6. CTRAS Treiber für Quattro und Sinix

CTRAS ist ein Netz-Treibersystem, das die Verbindung zu einem Quattro oder einem Sinix System erlaubt. CTRAS muß separat erworben werden, und steht nur in einer 16-bit Version zur Verfügung. CTRASX.DLL muß auf dem System vorhanden sein.
CTRAS erlaubt vollen Schreibzugriff auf BASIC-Dateisysteme. Die Beschreibungen der unten erwähnten Funktionen ist den entspr. CTRAS Handbüchern zu entnehmen. CTRAS bietet z.Zt. die einzige Möglichkeit der Verbindung zu Quattro Systemen. Betr. Sinix Systemen kann auch das Servermodul, das mit TRIO geliefert wird, benutzt werden.



55. Einstellung eines CTRAS Treibers

12.6.1. Name und Paßwort des Servers

Der Name gibt den benutzten Netzserver an. Das Paßwort gibt Zugang zu dem entspr. Netzserver. Name und Paßwort werden als Parameter 3 und 4 im NF_CONNECT Call weitergereicht.

12.6.2. Name und Paßwort des Anwenders

Dieses sind die Anwenderinformationen auf dem Server. Für ein Quattro System könnten der Name CMC, und das Paßwort MANAGER sein. Beide werden weitergereicht als Parameter 5 und 6 im NF_CONNECT Call.

12.6.3. Art der Verbindung

Um die CTRAS Funktionen zu erhalten, muß die Art der Verbindung auf NORMAL eingestellt sein.

56. CTRAS Optionen

12.6.4. Quattro/UNIX

Hier wird entschieden, welcher Server benutzt wird. Quattro entspricht 1, und benutzt die NF_CONNECT Funktionen in CTRAS. Sinix entspricht 2, und benutzt die NF_CONNECT Funktionen.

12.6.5. LU0 Path

Bei Sinix Systemen muß nur der LU0 Path als /XBASIC/0 angegeben werden. Dies gilt auch, wenn die Serverumgebung als $XBASIC.HOME festgelegt ist. Dies wird im NF_INIT Call benutzt.

12.7. X/BASIC Sinix C Interface Library Treiber

Diese Art ein X/BASIC zu betreiben erfordert kein CTRAS System, erlaubt aber vollen Schreibzugriff auf BASIC Dateien.
Die Netzverbindung wird mit dem TRIO UNIX Server aufrecht erhalten. Die Zugriffsfunktionen auf BASIC Dateien liegen in der C Interface Library, die kostenlos als nfmlib.a zusammen mit dem X/BASIC System geliefert wird.

57. Einstellung eines X/BASIC Library Treibers

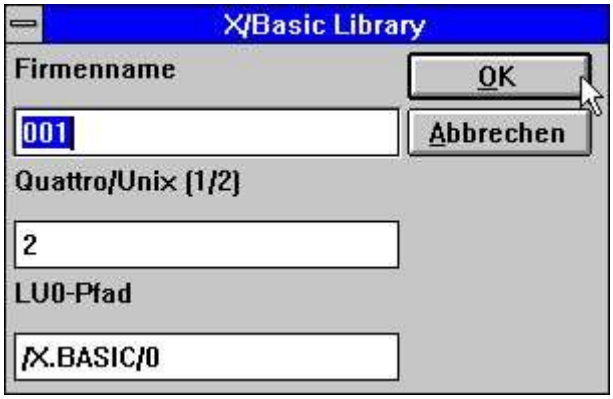
12.7.1. Art der Verbindung

Die Art der Verbindung muß TCP/IP sein.

ANMERKUNG: Auf einem SINIX System muß das Serverprogramm (SWTUSOCK) von einem X/BASIC Anwender gestartet sein.

12.7.2. Optionen

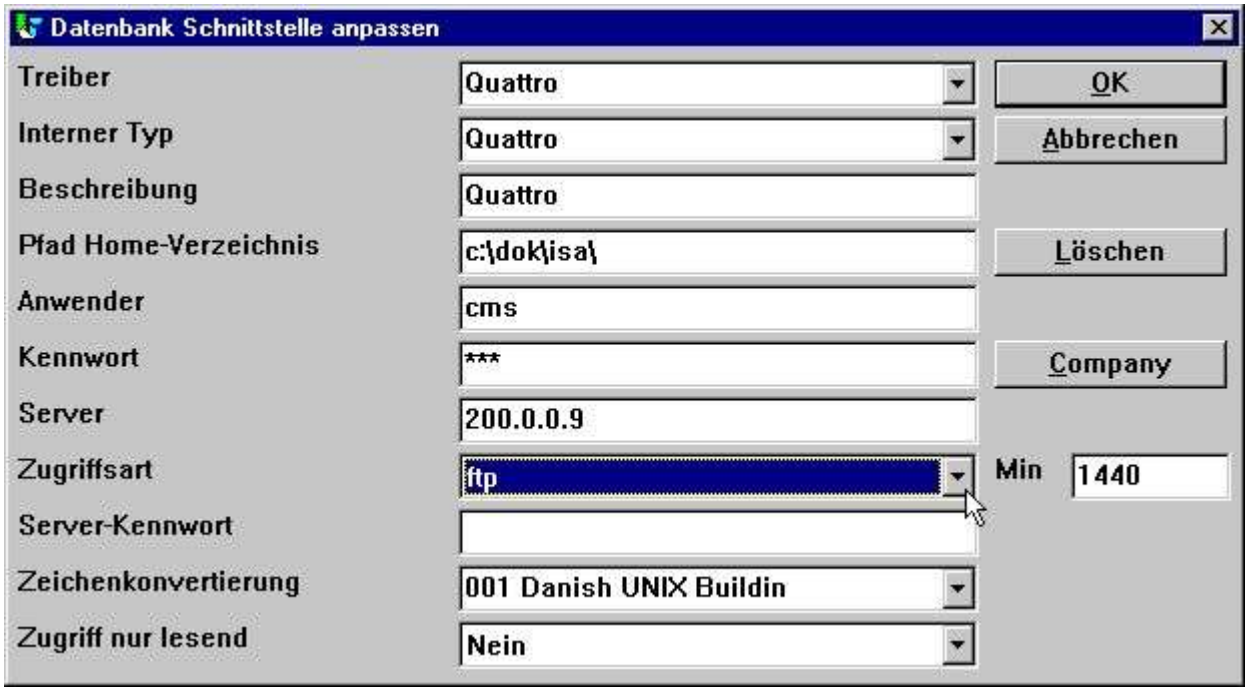
Die Optionen für diesen Treiber entsprechen denen des CTRAS Treibers.



58. X/BASIC Library Treiber Optionen

12.8. Quattro Schnittstelle mit 32 Bit FTP

Eine Alternative zu der vielleicht etwas langsamen CTRAS Schnittstelle kann den Quattro-Anwendern, die eine Netzwerkverbindung und den SSQ FTP Server benutzen, angeboten werden.
Die Schnittstelle benutzt FTP, um komplette Quattro Dateien zu übernehmen, anstelle des Lesens von einzelnen Sätzen. Nach der Übernahme wird ein lokales Lesen der Datei im Quattro Format durchgeführt. Um lange Dateinamen zu erlauben, empfehlen wir, die 32-Bit Version zu benutzen. Hiermit können Konflikte umgangen werden, die entstehen können, wenn Dateinamen in das Format xxxxxxxx.xxx für die 16-Bit Versionen umgewandelt werden sollen. Die Dateien werden kopiert, wenn diese von einem List-/IQ- Programm geöffnet oder von ODBC angesprochen werden. Die Update-Frequenz kann eingestellt werden für z.B.: immer, per Stunde, per Tag o.ä.



59. Einstellung von Quattro mit FTP

Aktiviert man die Zugriffsart FTP, muß auch ein Wert in das Feld MIN eingegeben werden. Hier gibt man die Anzahl Minuten zwischen dem FTP Transfers einer Datei an, oder mit anderen Worten, wie alt dürfen die Daten sein, bevor ein neuer Update erfolgt.
Im obigen Beispiel wurde 1440 eingegeben, also 24 * 60 Minuten, d.h. daß einmal täglich ein Update vorgenommen wird.
Gibt man 0 ein (oder leer) erfolgt der Update jede Stunde.

12.8.1. Offline Verarbeitung von Quattro Dateien

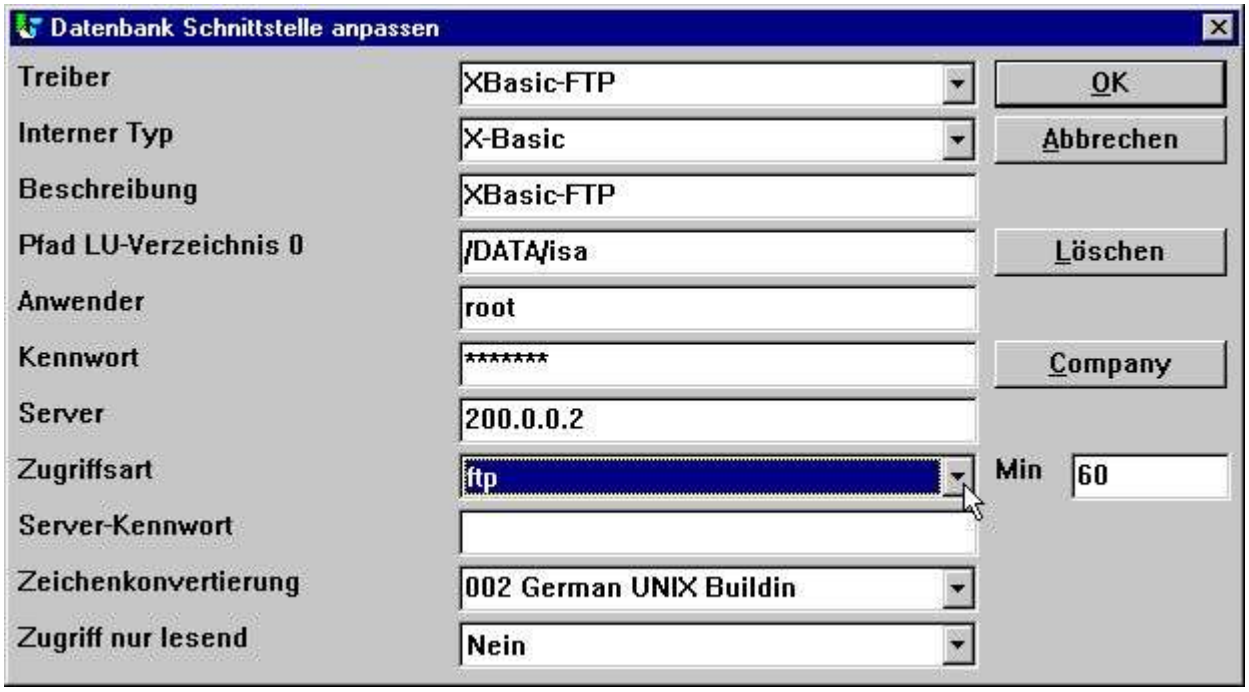
Gibt man den Wert -1 in MIN ein, erfolgt kein Datentransfer, und man arbeitet mit den Dateien, die man früher einmal von der Quattro erhalten hat.
Kann eine Datei aufgrund von fehlerhafter Verbindung nicht übertragen werden, wird eine entsprechende Fehlermitteilung ausgegeben. Steht eine früher übertragene Datei zur Verfügung, benutzt das Programm die Daten dieser Datei.
Es können auch Dateien manuell vom Quattro System auf den PC mit FTP oder Tapetransfer übertragen werden. Die Quattro Schnittstelle arbeitet mit binären Kopien der Quattro Dateien, einschl. Vorspann (Header Block).

12.8.2. Dateidefinitionen

Die Möglichkeit des direkten Imports von Quattro Dateidefinitionen besteht nicht. Die Dateidefinitionen müssen also auf eine andere Weise übernommen werden.

12.8.3. XBasic und andere Schnittstellen mit 32 Bit FTP

In der Praxis können Sie eine beliebige eingebaute Schnittstelle benutzen, um mit FTP Dateien zu übertragen und anschließend lokal zu verarbeiten. Die XBasic Schnittstelle auf einem RM System kann z.B. wie folgt eingestellt werden:

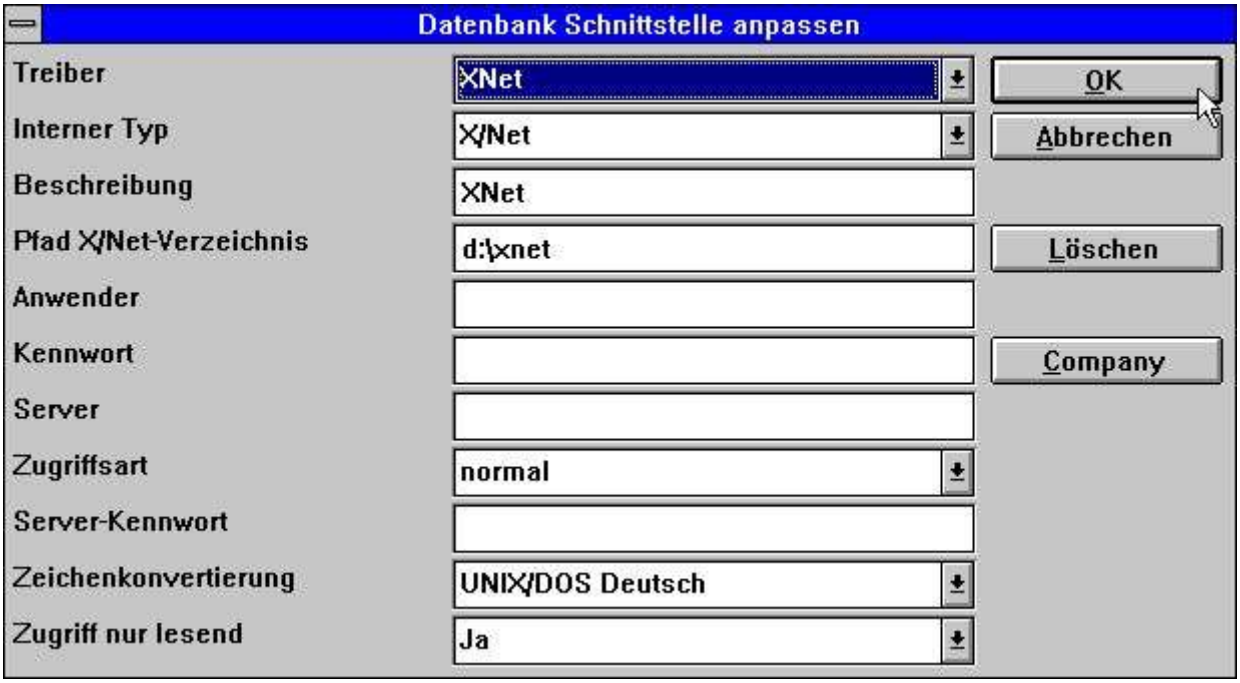


60. Einstellung von XBasic für FTP

Es steht jedoch nur ein Adressweg zur Verfügung, d.h. der lokale Adressweg wird identisch mit dem Adressweg auf dem Server. Der lokale Adressweg wird, falls nicht vorhanden, automatisch eingerichtet.
Beachten Sie Bitte, daß die XBasic Schnittstelle, auch in CTRAS Write Mode, auf einem UNIX System verwendet werden kann, auch wenn XBasic selbst nicht installiert ist.

12.9. XNet Treiber

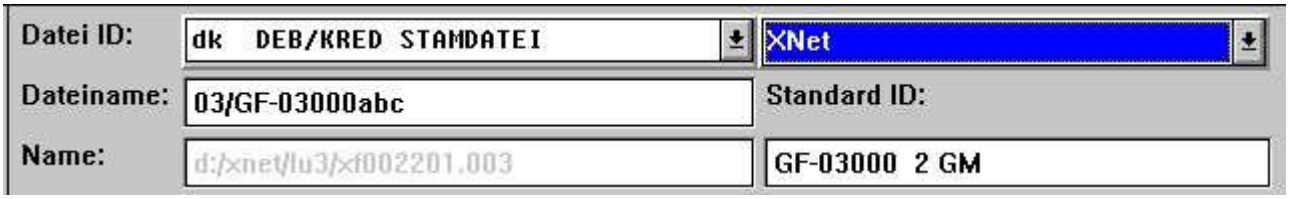
XNet ist ein BASIC System auf einem PC bzw. ein Netztreiber auf einem PC. Ein spezieller Server ist nicht erforderlich. Netzsofware, wie NOVELL oder Windows) muß installiert sein.
Schreibzugriffe auf Dateien sind auf nicht indizierte Dateien beschränkt. Datensätze können weder eingefügt noch gelöscht werden.



61. Einstellung des XNet Treibers

12.9.1. XNet Dateinamen

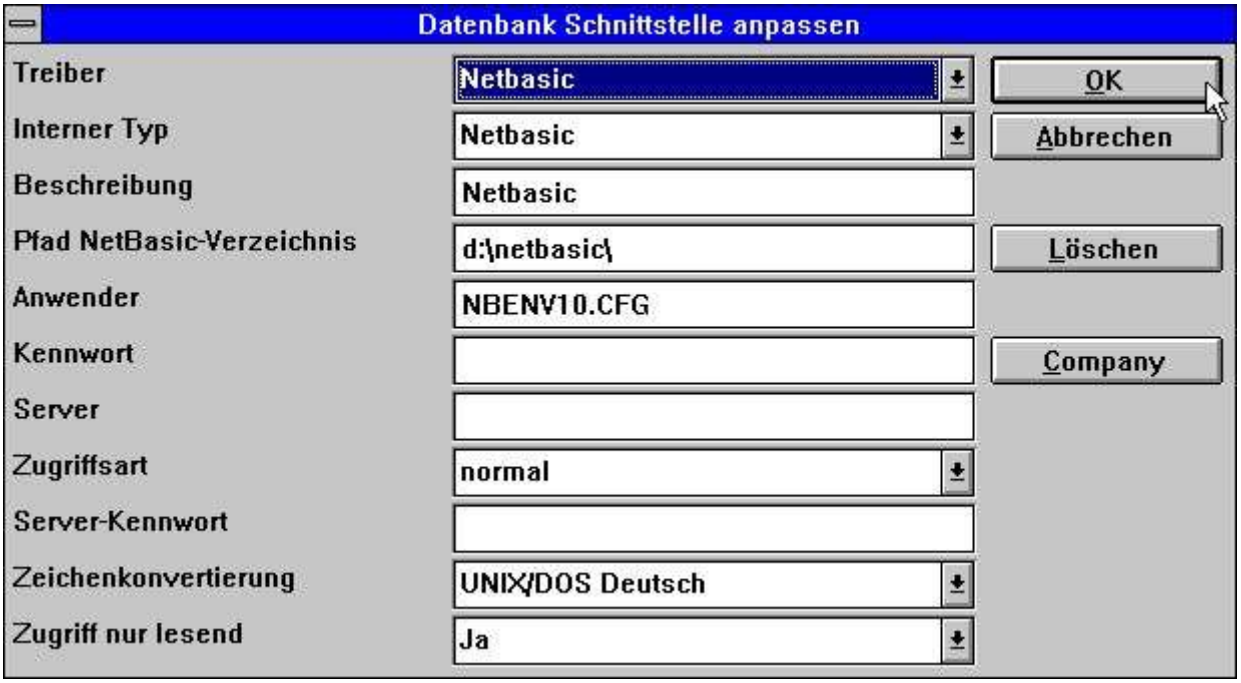
Unter XNet wird der BASIC Dateiname in einen DOS Dateinamen unter Zuhilfenahme einer INDEX-Datei, die im XNet Hauptdirectory eingetragen sein muß, umgewandelt.



62. XNet Dateinamen

12.10. Netbasic Treiber

Netbasic ist ein BASIC System, das auf einem PC installiert ist. Es ist kein spezieller Server erforderlich. Netzsoftware muß installiert sein.
Schreibzugriffe auf Dateien sind auf nicht indizierte Dateien beschränkt. Datensätze können weder eingefügt noch gelöscht werden.



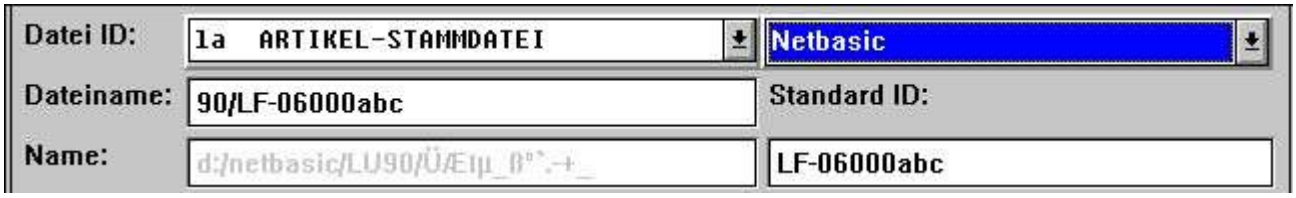
63. Einstellung eines Netbasic Treibers

12.10.1. Anwender

Dieses Feld kann den Namen der LU Konfigurationsdatei enthalten, wenn diese nicht NBENV3.CFG ist.

12.10.2. Netbasic Dateinamen

Die Struktur der Dateinamen ist weitgehend abhängig von der jeweiligen Version von Netbasic. Bitte prüfen Sie die Kompatibilität Ihrer aktuellen Version.
Alte Versionen verwendeten NBENV3.CFG und FILELIST.CFG zusammen mit den LU und Dateinamen. Die Version, die hier unterstützt wird, Version 4.25) benutzt NBENV10.CFG für die LU's, und einen internen Algorithmus, um den BASIC Dateinamen in einen nicht lesbaren DOS Dateinamen umzuwandeln.



64. Netbasic Dateinamen

12.10.3. NETBASIC für CTRAS Funktionen

Für NETBASIC kann eine CTRAS-gleiche DLL von Ihrem Netbasic-Lieferanten geliefert werden. Dies gibt Ihnen die Möglichkeit, Netbasic Dateien sowohl zu lesen als auch schreiben zu können.
Die DLL muß installiert werden, wenn Sie eine Netbasic Version 4.29 oder größer haben. Kleiner der Version 4.29 können Sie die in SW-Tools eingebaute Schnittstelle benutzen.
Verwenden Sie diese DLL, sollten Sie den Installationsanleitungen für CTRAS Schnittstellen folgen. Ferner müssen Sie eine gesonderte CTRAS (2013) Lizenz für TRIO haben.
Ungleich der Schnittstelle für Quattro/XBasic erlaubt die CTRAS Schnittstelle rückwärtiges Lesen von Netbasic Dateien.

12.11. UNIBASIC / Surfbasic / CX-Basic / Open-Basic Treiber

Die Struktur der UNIBASIC, Surfbasic und CX-Basic Dateien ist intern die gleiche. Ebenso sind die benutzten Treiber und die Einstellungsvorschriften gleich.
Schreibzugriffe auf Dateien sind auf nicht indizierte Dateien beschränkt. Datensätze können weder eingefügt noch gelöscht werden.

65. Einstellung eines UNIBASIC Treibers

12.11.1. UNIBASIC Dateien

Indexdateien sind in zwei Teile geteilt. Der Dateiname in Großbuchstaben stellt den Indexteil, in Kleinbuchstaben den Datenteil dar. Der Header der Datei bestimmt, ob es sich ein altes oder neues Indexsystem handelt. Beide Versionen werden unterstützt.
Benötigen Sie ein Testsystem auf einem PC, dann sollten Sie den Indexteil in die Datei xxxxxxxx.idx, und den Datenteil in die Datei xxxxxxxx.dat kopieren, da Dateinamen getrennt nach Groß-/Kleinschreibung nicht unterstützt wird.

12.12. NAVISION Financials

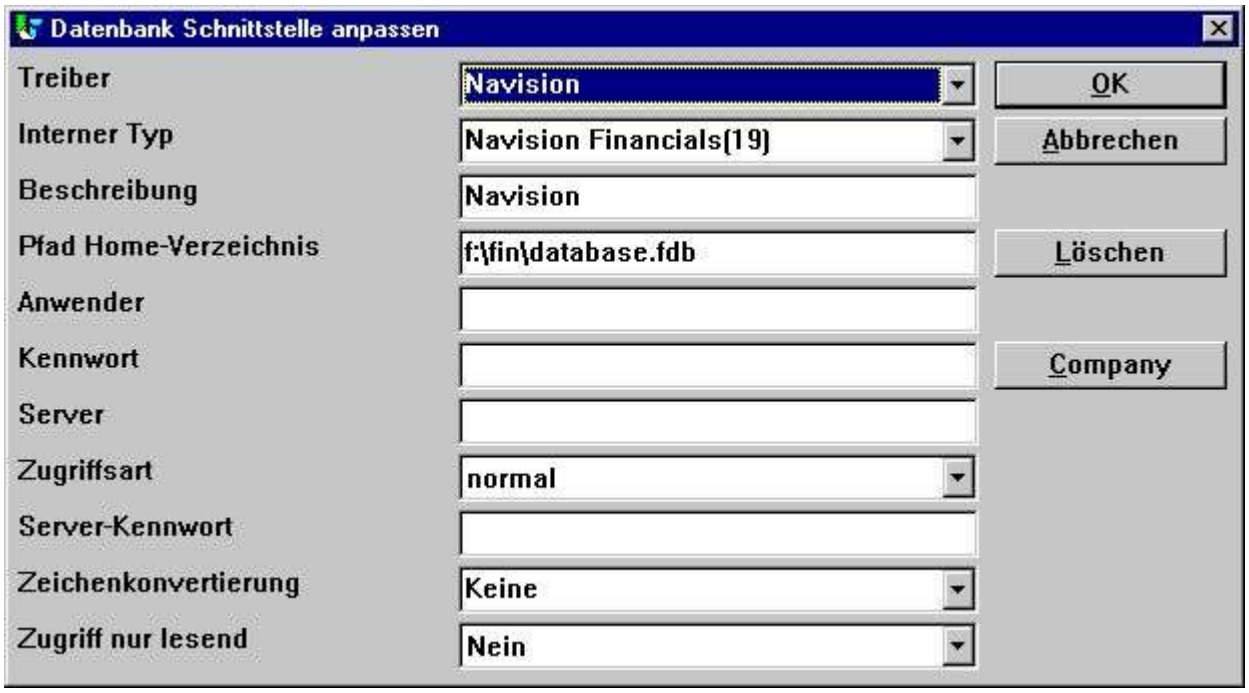
Die TRIO 32-Bit Version und der ODBC Treiber bieten jetzt über das C-Front Modul eine Schnittstelle zu NAVISION Financials. Das C-Front Modul muß auf dem Anwendersystem installiert sein. Lesen und Schreiben (Read/Write) sind implementiert. C/FRONT Version 1.2 oder größer werden unterstützt.

12.12.1. Installation von Navision Financials und C/FRONT

Installieren Sie NAVISION Financials und C/FRONT z.B. im Directory e:\fin. Versichern Sie sich, daß die 'path environment variable' das Directory e:\fin enthält. Ändern Sie diese Variable, muß das Windows System neu gestartet werden.

12.12.2. Installaion und Konfiguration der Navision Financials Schnittstelle

Folgen Sie der Anleitung für die Installation der ODBC Treiberschnittstelle in TRIO. Die Schnittstelle für NAVISION heißt 'Navision Financials'. Mit Hilfe der Funktion 'Datenbank Schnittstelle' können Sie 'Navision Financials' aufrufen. Die folgenden Einstellungen entsprechen der als Beispiel beschriebenen Installation.

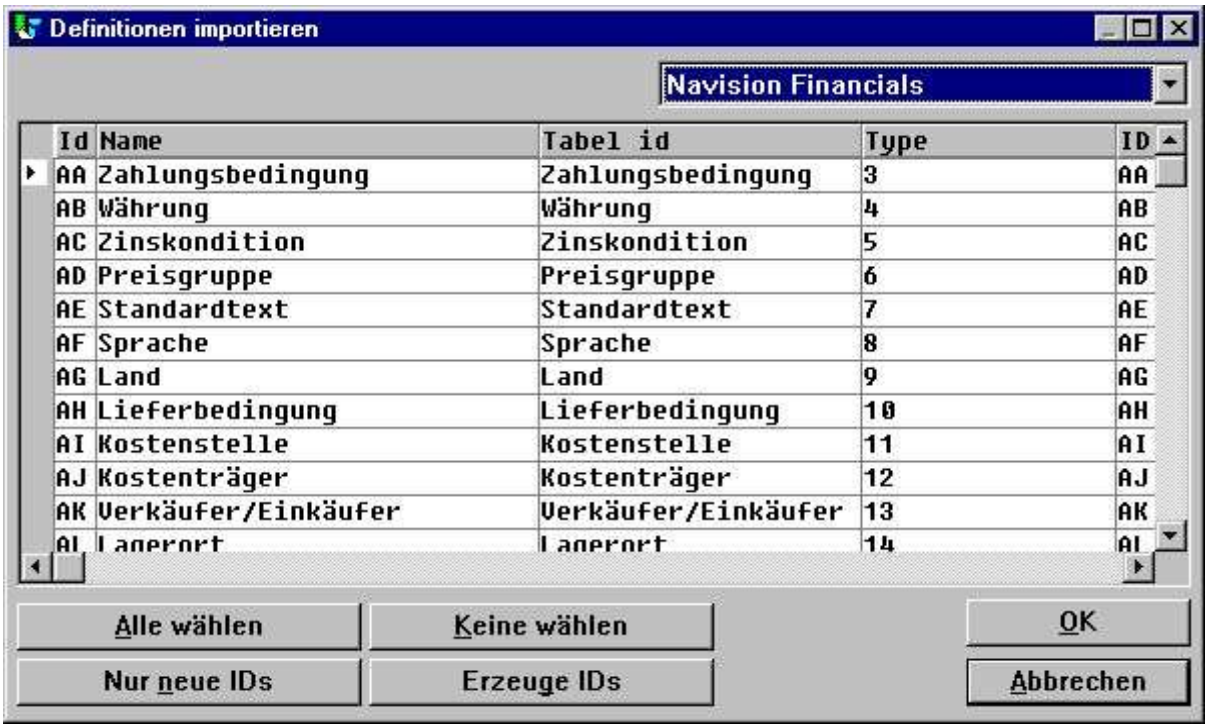


66. Beispiel: Einstellung der Navision Financials Schnittstelle

Benutzen Sie C/FRONT mit TCP/NETB als Serververbindung, wird das Feld 'Server' für die Eingabe 'server,typ' benutzt, wobei 'Server' der Servername und 'Typ' TCP oder NETB ist.

12.12.3. Import der Tabellendefinitionen

Im Menü 'Dateien' -> 'Import ODBC Definitionen...' können Sie die Schnittstelle 'Navision Financials' wählen, um die gewünschten (alle oder einige) Tabellendefinitionen zu importieren.



67. Import der Navision Tabellendefinitionen

12.13. CONCORDE C5/XAL

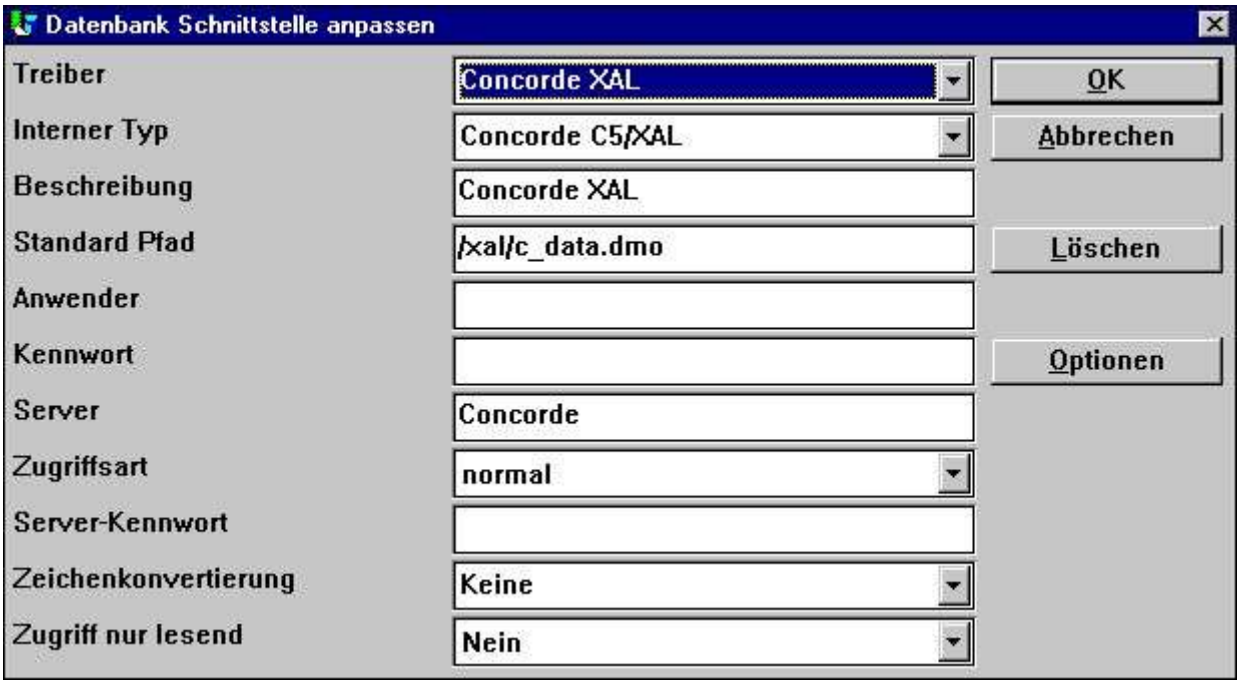
Eine Concorde C5/XAL Schnittstelle in 16-Bit oder 32-Bit Version als Read Only steht für TRIO und den ODBC Treiber zur Verfügung.

12.13.1. Export der Concorde Dateidefinitionen

Ist Concorde z.B. unter dem Directory e:\concorde installiert, müssen zuerst alle Dateidefinitionen in eine .DBD Datei übertragen werden.
Hierzu müssen Sie sich als 'Supervisor' anmelden (login) und das Menü 'Generell' -> 'Erweiterungen' -> 'Entwicklung' wählen. In diesem Menü wählen Sie 'Export' -> 'DBD' -> 'Alle + Enum'. Die exportierte Datei wird unter dem gleichen Directory wie Concorde gespeichert, also in diesem Fall e:\concorde.

12.13.2. Installation und Konfiguration der Schnittstelle zu Concorde

Folgen Sie der Anleitung für die Installation der ODBC Treiberschnittstelle in TRIO. Die Schnittstelle für CONCORDE heißt 'Concorde C4' oder 'Concorde C5/XAL'.
Mit Hilfe der Funktion 'Datenbank Schnittstelle' können Sie 'Concorde' aufrufen. Die folgenden Einstellungen entsprechen der als Beispiel beschriebenen Installation.



68. Beispiel: Einstellung der Schnittstelle zu Concorde

12.13.3. Import der Tabellendefinitionen

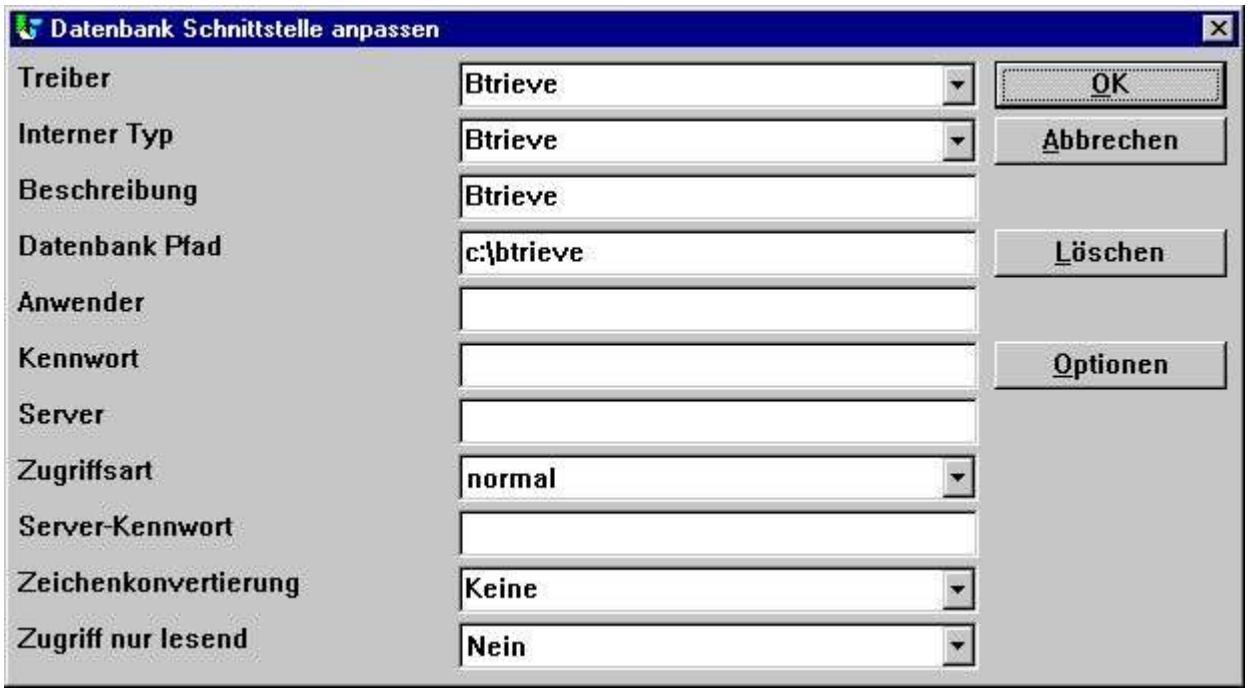
Aus dem Menü 'Datei' -> 'Import ODBC Definitionen...' wählen Sie die Schnittstelle 'Concorde', um die Tabellendefinitionen zu importieren.
Für den Import der Definitionen wird ein Anwendername gefordert. Der Anwendername ist 'BASIC'.
Nach Eingabe des Namens werden alle Tabellendefinitionen importiert. Es kann also keine Auswahl vorgenommen werden. Falls gewünscht, können anschließend eine oder mehrere der importierten Definitionen gelöscht werden.

12.14. Btrieve Version 5.10 und 6.15

Btrieve Dateien können direkt ohne Verwendung von ODBC angesprochen werden. Hierdurch erreicht man eine verbesserte Leistung, wenn man mit TRIO arbeitet.

12.14.1. Installation und Konfiguration der Schnittstelle zu Btrieve

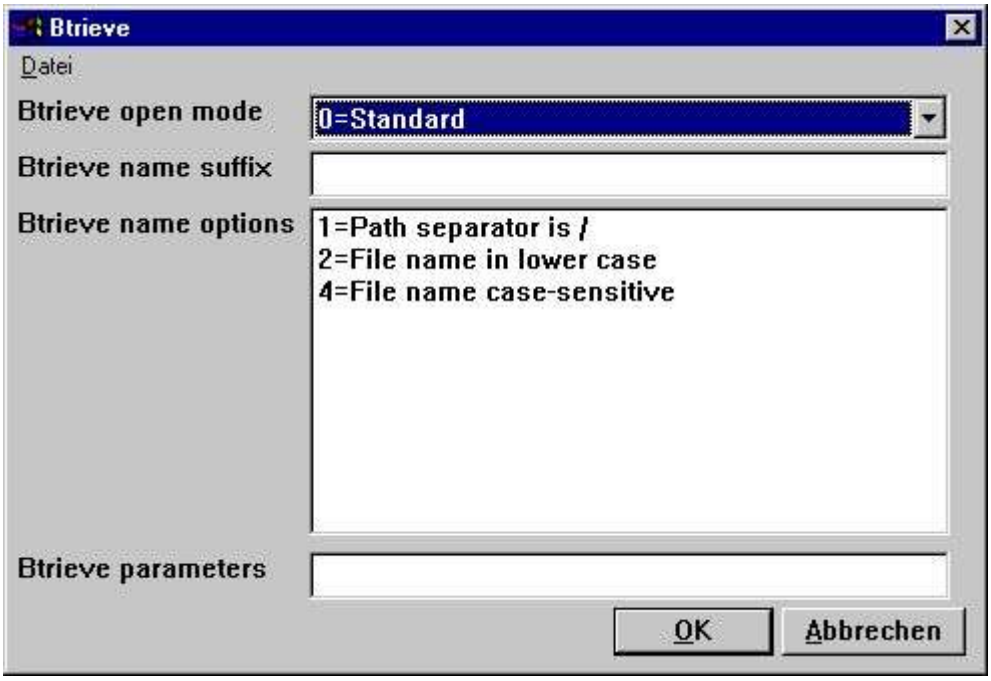
Folgen Sie der Anleitung für die Installation der ODBC Treiberschnittstelle in TRIO. Die Schnittstelle für Btrieve ist 'Btrieve'.
Mit Hilfe der Funktion 'Datenbank Schnittstelle' können Sie 'Btrieve' aufrufen. Die folgenden Einstellungen entsprechen der als Beispiel beschriebenen Installation.



69. Beispiel: Einstellung der Schnittstelle zu Btrieve

12.14.1.1. Erweiterte Optionen

Die Btrieve Schnittstelle bietet Ihnen die Möglichkeit, die Öffnungsart von Dateien und die Generierung von Dateinamen zu steuern.



70. Erweiterte Optionen für die Btrieve Schnittstelle

12.14.1.1.1. Öffnen einer Datei

Folgende Arten der Dateieröffnung stehen zur Verfügung:

- **Standard**
- **Exklusiv**
- **Single Engine File Sharing (nur Version 6.15)**
- **Multi Engine File Sharing (nur Version 6.15)**

.

Standard TRIO öffnet eine Btrieve Datei im Lesemodus (read only), es sei denn, daß in die entsprechende Datei von einer Liste oder einem DATAMASTER Programm geschrieben werden soll.

Wählt man die Art 'exklusiv', kann kein anderes Programm diese Datei ansprechen.

'Single..' und 'Multi Engine File Sharing' gelten nur für die Btrieve Version 6.15. Wir verweisen hier auf die entsprechende Btrieve Dokumentation für weitere Einzelheiten.

12.14.1.1.2. Suffix Dateiname

Das Suffix des Dateinamens einer Btrieve Datei wurde in TRIO Version <= 006.008 immer gleich .DAT gesetzt. Btrieve Dateien können jetzt ohne Suffix oder mit einem Suffix ungleich .DAT bestehen. Das gewünschte Suffix können Sie in einem dafür vorgesehenen Feld bei der Installation der Schnittstelle eingeben.

12.14.1.1.3. Optionen Dateiname

Die Optionen bestimmen, wie der Dateiname von TRIO generiert wird. Diese Optionen überschreiben die normal generierten Dateiadressen. Als Beispiel sei angenommen, daß die Schnittstelle mit dem Datenbank Adressweg

c:/btrieve/database

und dem Dateinamen

customer

eingerichtet wurde. Als Suffix wurde

dat

angegeben. Der aktuelle Adressweg für diese Datei ist also

c:/btrieve/database/customer.dat

Da Btrieve nicht die Verwendung von / in einem Adressweg erlaubt, werden standardmäßig alle / mit einem \ ersetzt. Dies gibt den neuen Adressweg

c:\btrieve\database\customer.dat

Sollten in neueren Btrieve Versionen Änderungen vorgenommen werden, kann dies hier geändert werden.

12.14.1.1.4. Parameter

Bei Aufruf von Btrieve Version 5.10 müssen eine Reihe von Startparametern an Btrieve weitergereicht werden. Diese Parameter sind standardmäßig wie folgt gesetzt:

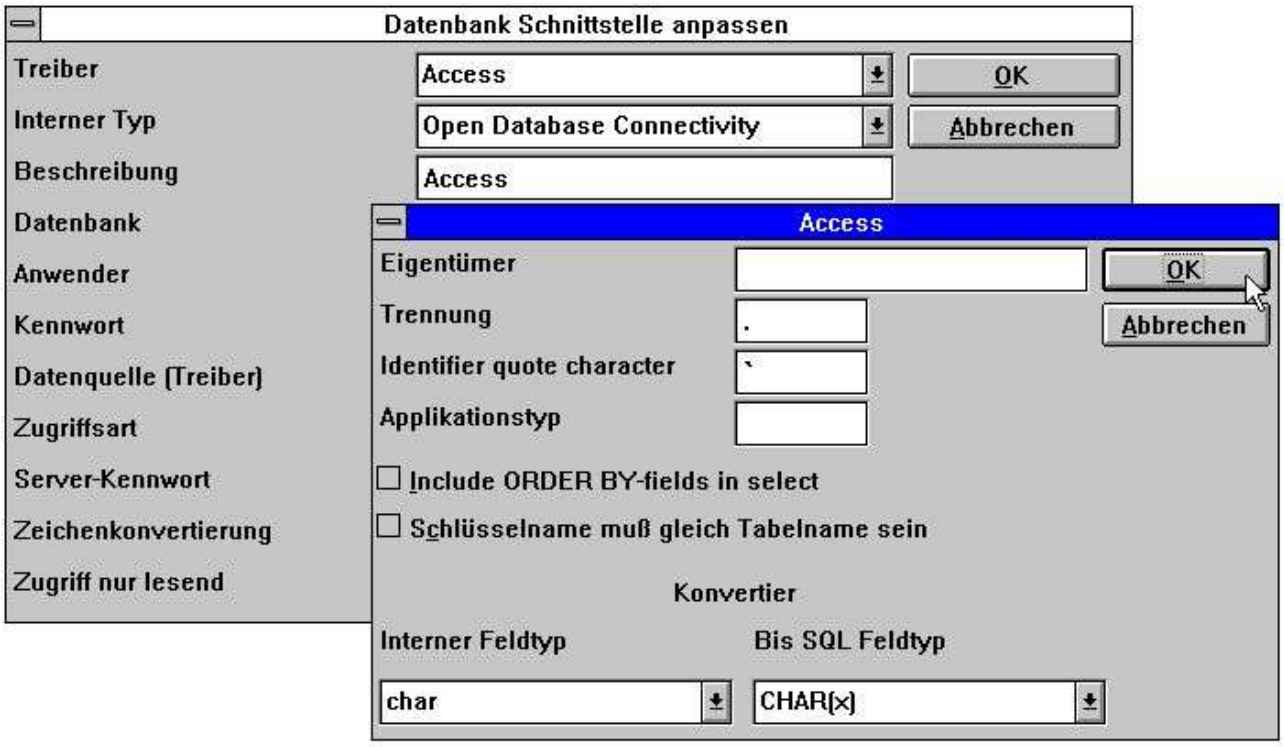
/m:48 /b:16 /f:20 /l:20 /p:4096

.

Für die detaillierte Beschreibung dieser Parameter verweisen wir auf das entsprechende Handbuch für Btrieve Version 5.10. Die Parameterkette hat für die Btrieve Version 6.15 (oder größer) keine Bedeutung.

12.15. ODBC Treiber Einrichtung

Da einige ODBC Treiber besondere Informationen verlangen, die nicht unmittelbar vom Treiber selbst generiert werden können, wurde das Einrichtungsfenster für ODBC-Treiber wie folgt erweitert:



71. Einrichten eines ODBC Treibers

Bitte beachten Sie die entsprechenden Handbücher der Treiberlieferanten, bevor Sie Änderungen in diesen Feldern vornehmen.

12.16. ODBC Treiberschnittstelle - Anwendungstyp

Der Typ der Anwendung wird für die besondere Funktionalität im Zusammenhang mit einer Dateischnittstelle benutzt. Folgende Typen werden verwendet:
Beachten Sie, daß durch Addition der Werte mehrere Typen gleichzeitig gesetzt werden können.

12.16.1. Type 1 - Unterstützung der Informix Datenbank Sperrung

Das Problem mit dem INFORMIX Treiber bestand darin, daß bei Abarbeiten eines normalen 'select....from....' die Tabelle gesperrt wurde. Dies führte dann dazu, daß auch normale ALX Anwendungen gesperrt waren. Die Lösung hierfür war ein spezielles SQL Kommando vor jedem 'select', welche von SNI im Zusammenhang mit ALX Anwendungen benutzt wird.
Damit SW-Tools TRIO hier eine Unterstützung anbieten kann, muß diese Kennung in der Datei <Trio Adressweg>\basis.ssv auf 1 gesetzt werden. Diese Änderung kann mit Hilfe der 'Sonderoption' für den Datenbanktreiber vorgenommen werden.

12.16.2. Type 4 - ORDER BY

Da nicht alle ODBC Treiber die Microsoft Standards unterstützen, können SQL Fehler auftreten, wenn eine Liste mit Zugriff auf eine Tabelle bei Benutzung des ODBC Treibers gedruckt werden soll. Die Fehlermitteilung gibt ein fehlendes ORDER BY für die Tabelle an.
Eine Liste, die ein Feld z.B. aus der Kundentabelle wählt, wobei der Kunde der primäre Schlüssel ist, generiert folgende Zeile:

SELECT kunde,name,addresse FROM customer_table

Wenn Typ 4 gesetzt ist, wird die Zeile wie folgt generiert:

SELECT kunde,name,addresse FROM customer_table ORDER BY kunde

12.16.3. Typ 4096 - Alphanum. Felder dürfen NICHT leer sein

Da einige ODBC Treiber / SQL Datenbanken nicht alphanumerische Felder ohne Inhalt, also "", unterstützen, muß TRIO mitgeteilt werden, daß alle alphanumerischen Felder einen Inhalt haben müssen.
Dieser Typ sorgt dafür, daß alle leeren alphanumerischen Felder mit einer Leerstelle gefüllt werden. Es wird der Packungstyp 1048 verwendet.

12.16.4. Typ 8192 - ODBC Treiber, die nur eine Tabellenverbindung unterstützen

Da einige ODBC Treiber / SQL Datenbanken eine eindeutige Verbindung zu jeder Tabelle fordern, versucht TRIO, diese Tabellenzugriffe zu optimieren (nur eine Verbindung zur Datenbank), kann es notwendig sein, diesen Typ zu setzen.
Ist dieser Typ gesetzt, wird TRIO eine Verbindung zu jeder benutzten Tabelle in der Datenbank einrichten. Achtung: Wenn der ODBC die Eingabe des Anwendernamens und Kennworts für die Verbindung zur Datenbank verlangt, muss diese für jede benutzte Tabelle eingegeben werden. Um dieses zu vermeiden, muß der Anwendername und das Kennwort in der Datenbankschnittstelle von TRIO angegeben werden.

12.17. ISAM Datenbankschnittstelle - Anwendungstyp

Der Anwendungstyp kann in ISAM auch für die Kontrolle der Firma benutzt werden.

12.17.1. Typ 2 - Unterstützung des SAMSON Systems

Da SIEMAX/SAMSON Systeme die Firmennummer als Teil des Index benutzen, muß im normalen TRIO System der Index als <4-stellige Firmennummer> gefolgt von der Kundennr. in einer IQ Anwendung eingegeben werden. Dieses Problem kann jetzt dadurch gelöst werden, daß man den Anwendungstyp für die benutzte C-ISAM Schnittstelle auf 2 setzt. In diesem Fall wird die Firmennummer automatisch eingesetzt (aus der Schnittstelle bzw. dem Untersystem übernommen).

13. Sonstige Treiber / Anmerkungen zu einigen Anwenderpaketen

Folgende weitere Treiber stehen auf Anforderung zur Verfügung:

- **Dataflex**
- **Digital Access Manager**
- **ISAM**
- **Speedbase**

.

13.1. GLOBAL 3000 Speedbase und Global 2000 ISAM/DMAM

TRIO unterstützt jetzt diese Systeme. Eine gesondert Anleitung hierfür ist auf Wunsch erhältlich.

13.2. ALX Datenbank

Die COMET/ALX Lösung bietet eine Informix Datenbank an, die mit dem installierten ODBC Treiber angesprochen werden kann.
Um Zugriff zum Data-Dictionary zu erhalten, müssen folgende Schritte ausgeführt werden:

1. **Installieren des erforderlichen Treibers, ODBC Treiber für Informix.**
2. **Setzen des Parameters 'Anwender', z.B. alxdemo**
3. **Setzen des Parameters 'Paßwort', z.B. alxdemo**
4. **Angabe des Servernamens, z.B. dbm216**
5. **Importieren der ODBC Tabellen**

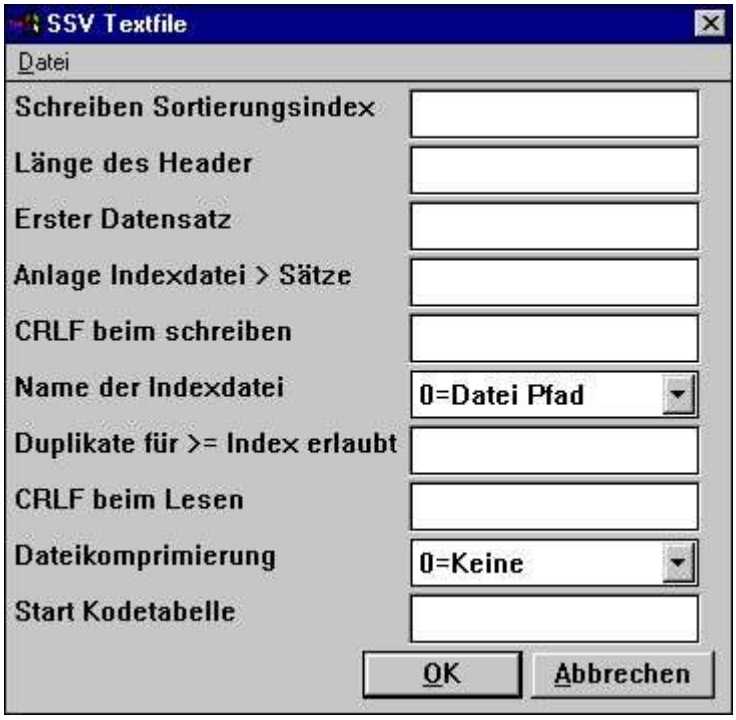
Wurden Tabellen importiert, kann jede Tabelle auf spezielle Firmen zugreifen. Hierzu muß die Firme als Teil des Tabellennamens eingegeben werden. Soll z.B. die Firma 999 aus der Tabelle ac2010 angesprochen werden, muß der Tabellenname

"c999".ac2010

lauten.

13.3. SSV Datenbankschnittstelle

Die SSV Schnittstelle für Semikolon getrennt Textdateien wurde ab TRIO (007.001) komplet überarbeitet. Gleichzeitig wurde die Verarbeitungsgeschwindigkeit und die Kompatibilität für Schlüsselstrukturen erhöht.
Komprimierte Dateien wurden eingeführt, um Platz sparen zu können.
Folgende erweiterte Optionen stehen für die SSV Schnittstelle zur Verfügung.



72. Erweiterte Optionen für den SSV Treiber

Beachten Sie, daß mit Bewegung des Mauscursors über ein Feld, ein Hilfstext für das entsprechende Feld angezeigt wird.

13.3.1. Schreiben Sortierungsindex

Bestimmt die Sortierung beim Schreiben einer Datei.

0=keine Sortierung
>0=Sortierung der Datei entsprechend dieser Indexdefinition.

13.3.2. Länge des Vorspanns (Header)

Nur von der SSV Schnittstelle benutzt. Bestimmt das erste gültige Datenbyte in einer Datei.

13.3.3. Erster Datensatz

Erster benutzter Datensatz in der Datei. Datensätze mit einer kleineren Satznummer werden ignoriert.
Erste Datenposition wird wie folgt berechnet:

Länge Vorspann + erster Datensatz * Satzlänge

Enthalten SSV Dateien Textzeilen, entspricht dies der Eingabe der ersten Textzeile in der Datei. Als Standardwert ist der Wert 1 eingesetzt, d.h. es wird Platz für den Vorspann reserviert. Soll die erste Zeile eingeschlossen werden, muß der Wert 0 angegeben werden.
Das Feld kann folgende Werte enthalten:

512 Dezimal
0x200 Hexadezimal
***8,B16P1006 Indirekte Berechnung**

Die indirekte Berechnung, gekennzeichnet durch ein vorangestelltes *, gibt in obigen Beispiel an, daß ein Feld mit dem Format 8 in der Datei ab exakter Byteposition 16 gelesen werden soll, und zwar unter Verwendung des Packungstyp 1006.

13.3.4. Anlage Indexdatei > Sätze

Bei großen SSV Dateien kann das Sortieren über Indexbegriffe längere Zeit in Anspruch nehmen.
Mit Hilfe dieses Parameters kann eine externe Indexdatei angelegt undimmer, wenn möglich, benutzt werden. Als Standardwert ist 10000 vorgegeben.
Solche Indexdateien können im TMP Directory zusammen mit der eigentlichen Datei abgelegt werden.

13.3.5. CRLF beim Schreiben

Beim Zurückschreiben einer SSV Datei werden normalerweise die gleichen Zeichen für CRLF (neue Zeile) geschrieben, die von der Datei gelesen wurden. Für eine neue Datei werden die Zeichen LF = 0x0A benutzt. Für hexadezimales Schreiben können folgende Zeichen für CRLF (neue Zeile) fest vorgegeben werden:

0x0a nur Zeilenschaltung (LF)
0x0d nur Wagenrücklauf (CR)
0x0a0d neue Zeile (LF+CR)
0x0d0a neue Zeile (CR+LF)

13.3.6. Name der Indexdatei

Der Adressweg und Name für die Indexdatei einer umfangreiche SSV-Datei ist von folgenden Parametern abhängig:

0 = Adressweg/Dateiname.Xnn
1 = TMP/Dateiname.Xnn

13.3.7. Duplikate für >= Index

Indexduplikate sind normalerweise nicht zugelassen.
Definiert man einen sog. Matchcode-Index, müssen gleiche Indizes selbst getrennt werden. Dies kann z.B. durch Hinzuaddieren der Satznummer erfolgen.
Man kann jedoch auch Indexduplikate für Indizes größer dem hier angegebenen Wert zulassen.
Bei Lesen einer SSV Datei sind Indexduplikate immer zugelassen, da der Aufbau einer Datei sehr unterschiedlich sein kann.

13.3.8. CRLF beim Lesen

Beim Lesen einer SSV Datei wird die Zeichenfolge für CRLF automatisch erkannt und beim Zurückschreiben wieder benutzt.
Man kann hier jedoch eine Zeichenfolge für hexadezimales Lesen vorgeben, Siehe CRLF beim Schreiben.

13.3.9. Dateikomprimierung

0 = Keine Datenkomprimierung (höchste Lesegeschwindigkeit für eine Datei)
1 = Datenkomprimierung über die Programme COMPRES/EXPAND (LZ Standard unter Windows)

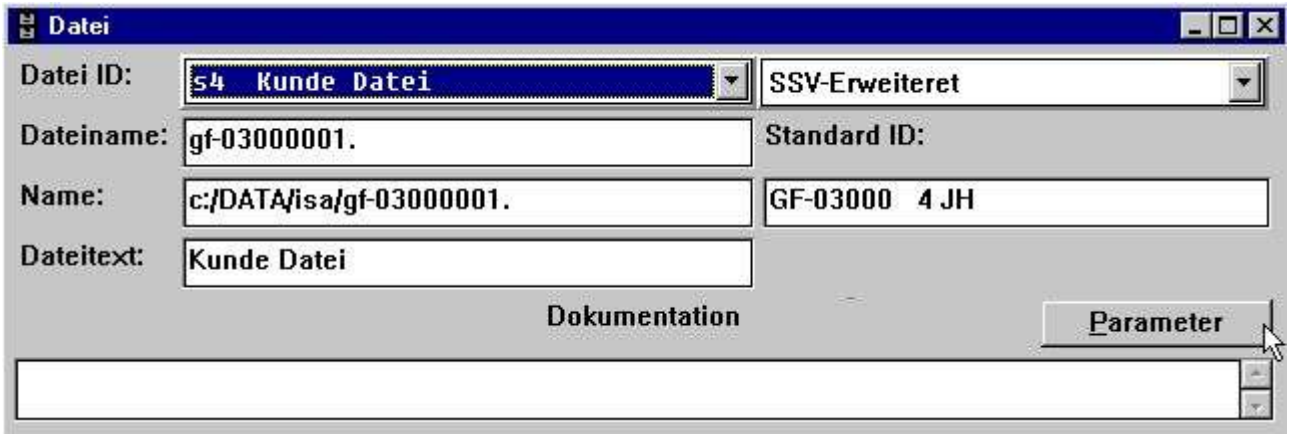
Bei 1 wird eine komprimierte Datei automatisch erkannt und wie jede andere Datei gelesen. Man muß jedoch selbst für die Komprimierung sorgen, da ein normales Schreiben immer unkomprimiert erfolgt.

13.3.10. Start Codetabelle

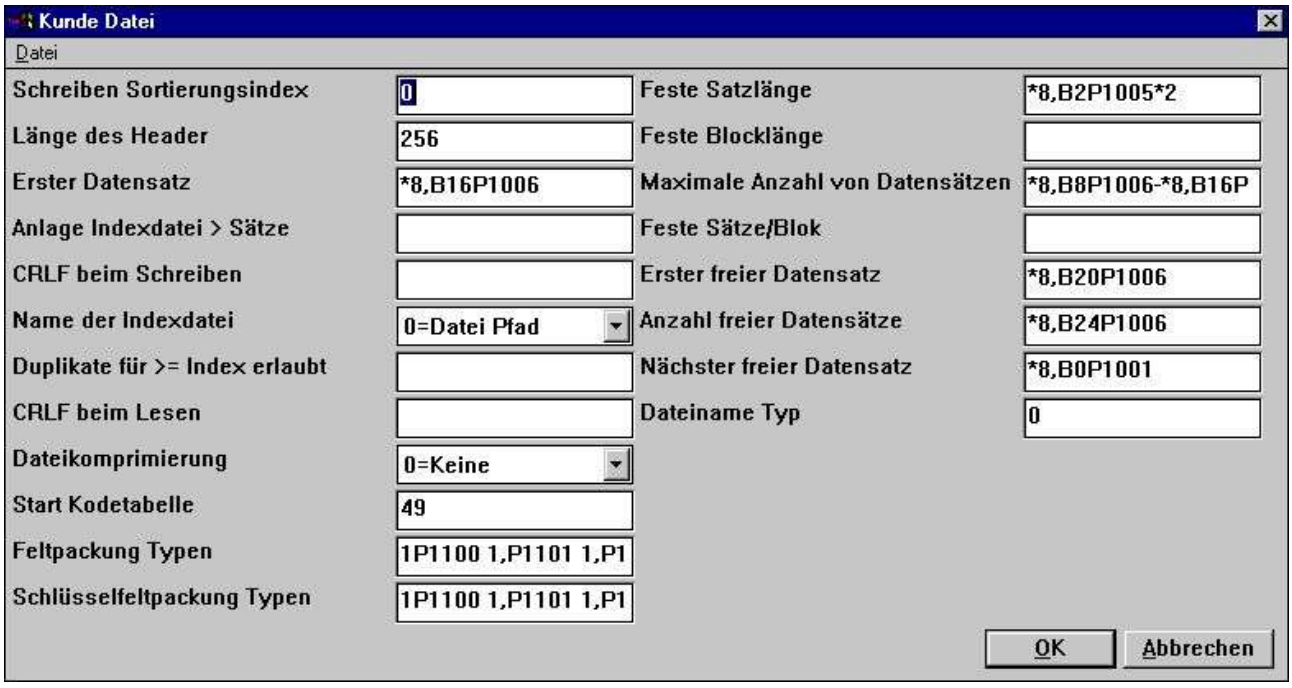
Die Codetabelle, die für die Schnittstelle definiert ist, wird auch für die Konvertierung der einzelnen alphanumerischen Felder einer Datei benutzt.
Gibt man hier eine Codetabellennummer an, können Sie die gesamte Datei unmittelbar nach Lesen bzw. vor Schreiben konvertieren, d.h. daß auch die Trennzeichen für Zeilenschaltung (LF) und Wagenrücklauf (CR) konvertiert werden.

13.4. Erweiterte SSV Schnittstelle

Erweiterte SSV Parameter für jede einzelne Datei stehen jetzt zur Verfügung, wobei nicht nur Textdateien, sondern auch andere einfache Dateien von dieser Schnittstelle behandelt werden können.
Auch komplexe Dateistrukturen, wie z.B. indizierte X-Basic Dateien mit fester Satzlänge, können für direkten Zugriff definiert werden, wenn der Indexteil übergangen wird. Die Einrichtung eines solchen Treibers verlangt jedoch tiefergehende Kenntnisse über das jeweilige Dateisystem.
Die erweiterten Optionen können bei der Treiberinstallation für jede einzelne Datei gesetzt werden. Parameterschalter wurden im Fenster für die Hauptdatei hinzugefügt.



73. Parameterschalter für einzelne Dateien



74. Erweiterte Parameter für den SSV Treiber, Definition einer X/Net Datei

13.4.1. Feste Satzlänge

Eine Eintragung in diesem Feld ändert eine SSV Datei von variabler in feste Satzlänge. Hierdurch kann die erweiterte SSV Schnittstelle auf eine Reihe unterschiedliche Dateien zugreifen.

***8,B2P1005*2**

definiert, daß ein Feld mit dem Format 8 in der Datei ab exakter Byteposition 2 gelesen werden soll, und zwar unter Verwendung des Packungstyp 1005, multipliziert mit 2.

13.4.2. Feste Blocklänge

Zusammen mit der festen Satzlänge kann eine Blocklänge vorgegeben werden, damit eine bestimmte Anzahl von Sätzen in einen Block zusammengefaßt werden kann und möglicherweise mit einem Zwischenraum zwischen den Blöcken.

***8,B2P1005*2**

definiert, daß ein Feld mit dem Format 8 in der Datei ab exakter Byteposition 2 gelesen werden soll, und zwar unter Verwendung des Packungstyp 1005, multipliziert mit 2.

13.4.2.1. Maximale Anzahl von Datensätzen

Eine SSV Datei kann auf eine maximale Anzahl von Datensätzen beschränkt werden.
Das Feld kann folgende Werte enthalten:

***8,B8P1006-*8,B16P1006**

definiert, daß die zwei Felder mit Format 8 in der Datei ab exakter Byteposition 8 und 16 gelesen werden sollen, und zwar unter Verwendung des Packungstyp 1006, wobei die Werte der Felder voneinander abgezogen werden sollen.

13.4.3. Feste Sätze/Block

Zusammen mit der festen Satzlänge kann eine Blocklänge vorgegeben werden, damit eine bestimmte Anzahl von Sätzen in einen Block zusammengefaßt werden kann und möglicherweise mit einem Zwischenraum zwischen den Blöcken.

13.4.4. Erstes freier Datensatz

Es kann hier der erste freie Datensatz für die SSV Schnittstelle angegeben werden. Hierduch erreicht man, daß unnötiges Lesen von freien Sätzen entsprechend der Kette für freie Datensätze vermieden wird.

***8,B20P1006**

definiert, daß ein Feld mit dem Format 8 in der Datei ab exakter Byteposition 20 gelesen werden soll, und zwar ungepackt unter Verwendung des Packungstyp 1006.

13.4.5. Anzahl freier Datensätze

Es kann hier die Anzahl freier Datensätze für die SSV Schnittstelle angegeben werden. Hierduch erreicht man, daß unnötiges Lesen von freien Sätzen entsprechend der Kette für freie Datensätze vermieden wird.

***8,B24P1006**

definiert, daß ein Feld mit dem Format 8 in der Datei ab exakter Byteposition 24 gelesen werden soll, und zwar ungepackt unter Verwendung des Packungstyp 1006.

13.4.6. Nächster freier Datensatz

Eine Kette freier Datensätze kann definiert werden, wobei jeder freie Satz auf den nächsten verweist. Hierduch erreicht man, daß unnötiges Lesen von freien Sätzen entsprechend der Kette für freie Datensätze vermieden wird.

***8,B0P1001**

definiert, daß ein Feld mit dem Format 8 in der Datei ab exakter Byteposition 0 gelesen werden soll, und zwar ungepackt unter Verwendung des Packungstyp 1001. Der berechnete Wert zeigt auf den nächsten freien Datensatz.

13.4.7. Dateiname Typ

Reserviert für späteren Gebrauch. Das Feld muß 0 enthalten.

13.4.8. Feldpackung Typen

Kann nur für Dateien mit einer festen Länge benutzt werden. Normale SSV Dateien benutzen immer SSV Packung.
Der Art des Packens kann für die Felder direkt angegeben werden, z.B. P1013, oder als Standardangabe für jeden Feldtyp, getrennt durch Leerstellen.

1,P1100 1,P1101 1,P1102 1,P1103 1,P1104 1,P1105 1,P1106 1,P1107

bewirkt normale BASIC Packung für Alpha, 1 Wort, 2 Wort, ..., Call 60 usw.
Die ersten 6 beziehen sich auf die Feldtypen 0-5, gefolgt von P0-9.

13.4.9. Schlüsselfeldpackung Typen

Wie 'Typ Feldpackung', jedoch benutzt für Felder, die in einem Schlüssel benutzt werden. Siehe oben.
Eine normale SSV Datei benutzt immer LDCHAR/CALL60, wenn ein Feld Teil eines Schlüssels ist.

Figuren

1. Lizenzinformationen 9
2. Data-Dictionary 10
3. Zugriff zu den Funktionen über Menüs oder Symbolleiste 11
4. Andere Knöpfe in der Symbolleiste 12
5. Generelle Menüs und Funktionen in der Symbolleiste 12
6. Funktionen 13
7. Wahl der Funktion für die Installation der Treiber 14
8. Installierte Treiber 15
9. Installation weiterer Treiber 15
10. Wahl der Treiber Schnittstellenfunktion 16
11. Ändern der Treiberparameter 17
12. Import von ODBC Definitionen 29
13. Standard Definitionen von der BASIC Version der SW-Tools Dateidefinitionen 36
14. PUT Tabellendefinitionen in eine Textdatei 43
15. Id für eine neue Tabelle 46
16. Wahl des Treibers für eine bestimmte Tabelle 49
17. Datenbankverknüpfungen 55
18. Löschen von Tabellen 56
19. Satzüberblick 57
20. Reservierte Worte 58
21. Feldbeschreibungen 59
22. Einfügen und Löschen von Feldern 59
23. VIEW Menü 88
24. SQL-Namen und SQL-Typen 91
25. Byte, C-Byte, C-Variable und BASIC Variable 93
26. Satzinhalt einer Tabelle 96
27. Hexadecimaler Dump eines Tabellensatzes 99
28. Übersicht über Hilfstexte 102
29. Bestätigung einer DATAMASTER Dateikopie 106
30. EDIT Menü 108
31. Editierung der Indizes 109
32. Einfügen und Löschen von Indizes 109
33. AF-05000/020 Auftragszeile, Index und Verknüpfungen 115
34. Beispiel eines Pseudoindex und Satznummer Definition 132
35. Zugriff auf Auftragseingänge, Tabelle mit SQL Namen 134
36. Laufende Online Feldbeschreibung 146
37. Ausgabe der Feldhilfsbeschreibung 147
38. Ausgabe der Feldprüfvorschriften 148
39. Ausdruck der Dokumentation 151
40. Wahl des Druckers 152
41. Optionen für den Ausdruck der Dokumentation 153
42. Beispiel einer Tabellendokumentation 154
43. Beispiel ein Inhaltverzeichnis 154
44. Grundeinstellungen 155
45. Subsystem Menü 178
46. Öffnen eines Subsystems 179
47. Anlegen eines Subsystems 180
48. Firmenwahl 191
49. Dateinamen für Firmen 196
50. Kodetabelle zum Lesen von BASIC-Dateien 199
51. 'Import ODBC Definitionen für einen BASIC Treiber 210
52. Einstellen eines X/BASIC Treibers 215

53. X.BASIC Dateiname 216
54. Standard Firmennummer 217
55. Einstellung eines CTRAS Treibers 218
56. CTRAS Optionen 221
57. Einstellung eines X/BASIC Library Treibers 224
58. X/BASIC Library Treiber Optionen 226
59. Einstellung von Quattro mit FTP 227
60. Einstellung von XBasic für FTP 230
61. Einstellung des XNet Treibers 231
62. XNet Dateinamen 232
63. Einstellung eines Netbasic Treibers 233
64. Netbasic Dateinamen 235
65. Einstellung eines UNIBASIC Treibers 237
66. Beispiel: Einstellung der Navision Financials Schnittstelle 241
67. Import der Navision Tabellendefinitionen 242
68. Beispiel: Einstellung der Schnittstelle zu Concorde 245
69. Beispiel: Einstellung der Schnittstelle zu Btrieve 248
70. Erweiterte Optionen für die Btrieve Schnittstelle 249
71. Einrichten eines ODBC Treibers 254
72. Erweiterte Optionen für den SSV Treiber 265
73. Parameterschalter für einzelne Dateien 276
74. Erweiterte Parameter für den SSV Treiber, Definition einer X/Net Datei 276

Index

A

Abspeicherung 62
Absteigende.....132
Access 16;24;44;79;257
Aktivieren.....160
ALX 251;259
Ansi.....204
Anwender..... 6;199;220;229;259
Anwenderinformationen 215
Anwendername 17;241;254
Arbeitsdirectory.....172
Arbeitsfeld.....62
Ascii 96
Ausdruck..... 146;148;157;282

B

BASIC-Datei .. 35;78;192;193;194;213;282
BASIC-Dateisysteme.....213
BASIC-Programm 70
BASIC-System 67
BASIC-Variable 90;206
BASIS 182;185;188
Betragsfelder 58
Bitnummer 66
Bits..... 64;128;209
BTrieve 3;44;45;78
Byte 65;66;79;84;88;282
Bytegrenzen 65
Bytenummer..... 65;83;84;102

C

C-byte 88
C-Byte 88;282
C-ISAM 3;44;256
CODnnn 194
COMET ... 39;49;110;111;144;161;205;259
COMET-Datei 39;49
Concorde.....78;238;239;240;241;283
Copyright 4
C-Struktur.....88
CTRAS
198;209;213;216;217;219;221;222;225;231;283
CTRASX 209;213
C-Variable 88;89;282
C-Variablentype 89
CX-Basic 79;209;232

D

Datadict 185;187
Dataflex 78;195;257
DATAMASTER
72;73;74;101;102;121;143;145;171;245;282
Datasyst..... 185;187
Dateieröffnung 178;245
Dateiextension 200
Dateiformat 199
Dateiname
35;192;193;200;211;227;233;246;247;266;279;283
Dateisystem..... 206;207;210;271
Dateityp 13;14;45;182
Dateiübersichten 152
Dateiverknüpfungen..... 37;39;134;137;138
Datenbankfenster 141
Datenbankinterface..... 187;188
Datenbanksystem..... 77;113;199
Datenbanktreiber 78;86;121;251
Datenbanktyp 40;182
Datenquelle 18
DATUM..... 78
Datumfeld 57;163;169
DBQ..... 16
Demo-Dateien..... 10
Dezimalstellen.....57;59;206;207
Digital 257
Directory
107;121;122;126;152;154;172;181;185;235;239;264
DLL.....209;213;231
Dmsystem 185;189
DOS..... 12;19;163;193;227;230
Druckausgabe 154
Drucker.....147
Dump..... 94;95;96;282

E

EDIT-Menü 86;142
Einfügen..... 54;104;282
Eingabeprüfungen 143
Excel.....24;44

F

Feldarten..... 59
Feldbeschreibungen 54;141;282
Felddefinition 91
Feldformat
..... 57;59;80;84;87;156;201;206;207
Feldhilfe 142
Feldlänge..... 85
Feldname 56
Feldprüfung 74
Feldprüfvorschriften..... 143;282
Feldtyp.....67;280
Feldübersicht 141
Festplatte 194
FILELIST 230

FILES 152
Firmanummer 21;178;185
Firmenname 190;191
Float 78
Formatangabe 58
G
GET-Funktion 38
Grundeinstellungen 150;282
H
Hauptdirectory 227
Hauptindex 3
Header 223;233;262
HELP 142
Hexadezimal 263
Hilfe
31;37;38;51;54;109;132;172;236;240;243;251;264
Hilfstexte 97;103;282
HOME 218
HOST-Datei 18
I
Import
23;24;25;27;28;34;205;237;241;282;283
Indexbeschreibung 202
INDEX-Datei 227
Indexdateien 233;264
Indexdefinition 125;128;134;202;261
Indexfelder 72;73;210
Indexfolge 121;132
Indexname 105
Indexnummer 107
Indexsystem 233
Indextabellen 198
Indexteil 233;271
Indexwerte 116;120
Informix 251;259
INIT-Datei 151
IP-Adresse 18
IQ-Programme 54;159;172;197
Isam 78;79;109
ISAM 116;255;257;258
ISO8850 ..... 19;195
K
Kennwort 17;254
KNIR 120;122
Kodefeldern 143
Kodetabelle . 68;133;194;196;199;204;282
KOM 191
Kompatibilität 73;74;230;260
Konfigurationsdatei 229
L
Listgenerator 141;143;162
Lizenzvereinbarung 4
LOAD 37;39;49
Löschen 22;51;54;104;175;282
LU 192;211;218;229;230
LU0 ..... 211;218
LU0-Path 211
M
MANAGER 215
Markieren 41
Master 37;39;49
Microsoft 123;252
N
NBENV10 230
NBENV3 ..... 229;230
Netbasic 209;228;230;231;283
Netzserver 214
NF_CONNECT 214;215;217
NF_INIT 218
NOVELL 226
O
ODBC
3;16;17;18;19;21;23;24;28;44;45;53;75;76;78;86;87;102;109;129;145;195;196;203;204;205;222;234;236;237;238;240;241;242;243;249;250;252;253;254;259;282;283
ODBC-Tabellen 86
ODBC-Treiber 18;21;24;86;249
OemToAnsi 19;195
Online-Hilfe 7;141
P
Packen 113;201;208
Packtypen 78
Paßwort 17;18;177;205;214;215;259
Preferencemenü 172
Primärindex 107
Prüfvorschriften 98;103;143
PUT-Funktion 37;38
Q
Quattro
209;213;215;217;222;223;224;231;283
R
RAPGEN 141;174;184;196
Rechtsbündig 79
S
Satzinhalt 91;282
Satzlänge 63;64;263;271;272;273;275
Satznummer
114;115;116;122;127;136;263;267;282
Satztyp 39;49;111
Satzüberblick 52;93;282
Schlüssel
.. 72;109;116;117;118;119;120;252;281
Schlüsselbegriff 116;118

Schlüsseldefinition 108;109;115
Schlüsselfelder 69;82
SELECT 131;252
Server
18;46;126;182;192;193;199;205;215;217;219;222;225;226;228;236
Server-Kennwort 18
Servername 18;236
Serverprogramm 220
SID.....37;39;49
Sinix 213;217;218;219
SINIX.....220
Skalierung 50
SNI.....251
Socket 18;210
Sortierung 121;138;202;261
Spool 154
Sprache 155;194
SQL
75;76;86;87;100;109;129;130;145;251;252;253;254;282
SQL-Datenbank.....130
SQL-Name.....86;130;145;282
SQL-Namen 86;130;145;282
SQL-Typen 86;87;282
SQL-Wörter 100
SSD.....152
SSV
10;126;152;182;185;199;200;260;262;263;264;265;266;267;268;271;272;274;276;277;280;281;283
SSV-Datei..... 152;266
Standard-ID 37
Stopzeichen 70;102
Subsystem
172;173;174;175;179;181;184;185;186;187;189;191;282
Subsystemmenü.....191
Subsystemtype 186;187;188;189;190
Surfbasic 209;232
SURFBASIC.....44
SWTUSOCK.....220
Systemdateien 197
Systemfelder 43

T

TCP..... 18;19;210;220;236
TEMP 154
Testsystem 233
Textdatei 31;33;38;39;199;282
TIME 75;79
TIMESTAMP 75
TMP 154;264;266
Treibertyp 25;32;52;92

U

UNIBASIC..... 44;79;206;232;233;283
UNIX..... 12;18;19;195;210;217;219;225
UPDATE..... 76

V

Variablen..... 67
Variablentyp 206
Verbindung
18;109;135;136;143;213;216;220;223;254
Verbindungsart 18
Verknüpfung
104;108;111;134;136;137;138;139;140
Verknüpfungsdefinitionen 134

W

Wort 67;206;280
Wortgrenzen 65
Write 209;225;234
WW 43

X

XAL..... 238;240
X-Basic .. 44;46;161;182;193;209;210;271
XNet 182;226;227;283

Z

Zeichenkonvertierung 12;19
Zugriffsart 18;222