



Dansk Brugerhåndbog

Copyright © (1990-2022) SW-Tools ApS
Duevej 23
DK-3250 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

Contents

Contents	2
1. Preface.....	8
1.1. Licence information	9
1.2. Start of the Data-Dictionary program	10
1.3. About the user interface	11
1.3.1. Functions for table definition and amendment.....	13
2. Driver installation	14
2.1. Installation of drivers	15
3. Database interface.....	16
3.1. Database interface parameters	17
3.1.1. Type.....	18
3.1.2. Internal type	19
3.1.3. Description.....	20
3.1.4. Database	21
3.1.5. User name and password for ODBC databases	22
3.1.6. Data source, Connection type and Server password.....	23
3.1.7. Code conversion - Language dependent and special characters	24
3.1.8. Read only.....	25
3.2. Advanced driver options	26
3.3. Deleting an installed driver.....	27
4. Importing ODBC definitions.....	28
4.1. Importing ODBC table definitions into the Data-Dictionary	29
4.1.1. Driver type	30
4.1.2. Id.....	31
4.1.3. Table name	32
4.1.4. Table-id	33
4.1.5. Suggested id	34
5. Standard definitions.....	35
5.1. Loading standard table definitions from a textfile	36
5.1.1. Driver type	37
5.1.2. Id.....	38
5.1.3. Table name	39
5.1.4. Filename.....	40
5.1.5. Suggested id	41
5.2. Loading standard file connections	42
5.3. Put of table definitions to a textfile.....	43
5.4. Put of file connections to a textfile.	44
6. Table information	45
6.1. Creating a new table	46
6.2. Table parameters.....	47
6.2.1. Id.....	48
6.2.2. Type.....	49
6.2.3. Name	50
6.2.4. Real name.....	51
6.2.5. Text	52
6.2.6. Documentation	53
6.2.7. Standard id	54
6.3. Database relations enquiry.....	55
6.4. Deleting tables	56
6.5. Record overview	57
6.6. Reserved words	58
7. Field descriptions.....	59

7.1. Number	60
7.2. Name	61
7.3. Format	62
7.3.1. Editing of numeric fields.....	63
7.3.2. Floating decimals	64
7.3.3. (nnn) Table fields.....	65
7.3.3.1. (nnn+) Grouped table fields.....	66
7.3.4. W Work fields	67
7.3.5. (nn) Explicit record length.....	68
7.3.5.1. (nn,x) Explicit record length in bits	69
7.3.6. Bnn Explicit byte number	70
7.3.6.1. Bnnn,x Explicit bitnumber.....	71
7.3.7. Tn Fixing field type.....	72
7.3.8. Unn Specific codetable for single fields.....	73
7.3.9. R Right justified display	74
7.3.10. S Stop (delimiter) character in alphanumeric field	75
7.3.11. Fnn System field addressing	76
7.3.12. K,D - DATAMASTER Index fields	77
7.3.12.1. E,I,V,X,Z - DATAMASTER Special index fields	78
7.3.13. Cnn - DATAMASTER check for field	79
7.3.14. Qnnn Explicit SQL type.....	80
7.3.15. An Access mode	81
7.3.16. Pnn Packed fields	82
7.3.16.1. Pnnnn Explicit specification of the pack type.....	83
7.3.17. Vxxx Secondary packtypes	85
7.3.17.1. Secondary packtype parameters.....	86
7.3.17.2. Default secondary packing for BASIC dates.....	87
7.4. Information on the field line, the VIEW menu	88
7.4.1. Byte number	89
7.4.2. Field Layout.....	90
7.4.3. SQL-names	91
7.4.4. SQL-types	92
7.4.5. C-byte	93
7.4.6. C-Variable.....	94
7.4.7. Basic-variable.....	95
7.4.8. Record.....	96
7.4.9. Next record	97
7.4.10. Record overview.....	98
7.4.11. Hexadecimal dump of a record	99
7.4.12. Decimal dump of a record.....	100
7.4.13. Ascii dump of a record	101
7.4.14. Help overview	102
7.4.15. Check overview.....	103
7.4.16. Special text overview	104
7.4.17. Reserved words.....	105
7.5. Amending DATAMASTER files using the FDF module	106
7.5.1. DATAMASTER file conversion from BASIC to ODBC.....	107
8. Index, help and check definitions.....	108
8.1. Index definitions and table connections	109
8.1.1. Indexname.....	110
8.1.2. File.....	111
8.1.3. Directory.....	112
8.1.4. Key definition	113
8.2. Index considerations	114
8.2.1. Basic COMET AF-05000 example	115

8.2.2. Constants and selection	116
8.2.2.1. Multiple constants in index definitions	117
8.2.3. Packing	118
8.2.4. Record numbers.....	119
8.2.4.1. Index 0	120
8.2.5. Duplicates allowed	121
8.2.6. Zero suppress.....	122
8.2.7. Space fill.....	123
8.2.8. Part of fields.....	124
8.2.9. Key not in record (KNIR)	125
8.2.10. Index key definitions, LOGICAL index.....	126
8.2.11. Directory 00 recordnumber reading	127
8.2.12. ACCESS using SWODBC on non-indexed files.....	128
8.2.13. Suppressing IQ search on specific index (*xx)	129
8.2.14. Suppressing index update for a specific index (-xx)	130
8.2.15. Creating a pseudoindex for a file (+xx)	131
8.2.16. Index example	132
8.2.17. Key options, Descending index.....	133
8.2.18. ODBC Access orders example.....	134
8.2.19. Index SQL-name	135
8.2.20. Born indexes and ORDER BY	136
8.2.21. Descending indexes	137
8.2.22. Special code table	138
8.3. Table connection considerations.....	139
8.3.1. Variable length keys	140
8.3.2. X* Fixed parameter file connections	141
8.3.3. File connection definitions	142
8.3.4. Multiple connections to the same file using upper/lower case.....	143
8.3.5. Multiple connections to the same file using logical index.....	144
8.3.6. Forcing connections by READ	145
8.4. Field help descriptions	146
8.4.1. Editing field help description	147
8.5. Field check codes	148
8.6. Field special text	149
8.7. Make standard SQL-names.....	150
9. Printing documentation.....	151
9.1. Printer	152
9.2. Options.....	153
10. Preferences	155
10.1. Directories	156
10.1.1. File definitions.....	157
10.1.2. Database.....	158
10.1.3. Spool directories	159
10.2. Language	160
10.3. Sign	161
10.4. Date output format.....	162
10.5. Separators	163
10.6. Ask for save when changes in report	164
10.7. Internal date calculation format	165
10.7.1. Year 2000	166
10.7.1.1. Date sorting.....	167
10.7.1.2. DOS dates	168
10.7.1.3. Year 2001 problem	169
10.7.1.4. Year 2005 problem	170
10.7.1.5. Year 2010 problem	171

10.7.1.6. Year 2100 problem	172
10.7.1.7. Date input validation	173
10.7.1.7.1. IQ selection date input	174
10.7.1.7.2. CHAIN function masked date input.....	175
10.8. FDF Start parameters	176
11. Subsystems	177
11.1. The subsystem menu.....	178
11.2. Opening of a subsystem.....	179
11.3. Creating a subsystem	180
11.3.1. Name.....	181
11.3.2. Password.....	182
11.3.3. Company.....	183
11.3.4. File definition path.....	184
11.3.5. Database standard path	185
11.3.6. Reports directory	186
11.3.7. Databaselock	187
11.3.8. Description	188
11.3.9. Module.....	189
11.3.10. Type	190
11.3.10.1. Company subsystem type	191
11.3.10.2. Datadict,Reports and Datasyst subsystem types	192
11.3.10.3. Basis subsystem type	193
11.3.10.4. Dmsystem subsystem type	194
11.3.10.5. Standard subsystem type	195
11.4. Company filenames	196
11.4.1. Company filename search.....	197
11.4.2. DOS filenames for BASIC files	198
11.5. Codetables	199
11.5.1. Codetable numbers.....	200
11.5.2. The SORT codetable	201
11.6. System files	202
12. Database drivers.....	203
12.1. SSV-Textfile driver	204
12.1.1. Table name	205
12.1.2. Field format.....	206
12.1.3. Index description.....	207
12.2. ODBC Drivers	208
12.2.1. Codetable.....	209
12.3. BASIC drivers in general	210
12.3.1. Field format.....	211
12.3.2. Packed fields.....	212
12.3.3. Index descriptions	213
12.4. BASIC Interfaces.....	214
12.5. X/Basic driver	215
12.5.1. Table name	216
12.5.2. Company number.....	217
12.6. CTRAS driver for Quattro or Sinix.....	218
12.6.1. Servername and Server password	219
12.6.2. User and User password	220
12.6.3. Connection type	221
12.6.4. Quattro/Unix.....	222
12.6.5. LU0 Path	223
12.7. X/Basic Sinix C Interface Library driver	224
12.7.1. Connection type	225
12.7.2. Options	226

12.8. Quattro interface using 32 Bit FTP	227
12.8.1. Working offline with the Quattro files	228
12.8.2. File definitions.....	229
12.8.3. XBasic and other interfaces using 32 Bit FTP	230
12.9. XNet driver.....	231
12.9.1. XNet filenames.....	232
12.10. Netbasic driver.....	233
12.10.1. User.....	234
12.10.2. Netbasic filenames.....	235
12.10.3. NETBASIC using CTRAS functions	236
12.11. Unibasic / Surfbasic / CX-Basic / Open-Basic driver	237
12.11.1. Unibasic files.....	238
12.12. NAVISION Financials.....	239
12.12.1. Install Navision Financials and C/FRONT	240
12.12.2. Install and configure the Navision Financials interface	241
12.12.3. Import of table definitions	242
12.13. CONCORDE C5/XAL	243
12.13.1. Exporting the Concorde file definitions.....	244
12.13.2. Install and configure interface to Concorde	245
12.13.3. Import of table definitions	246
12.14. Btrieve version 5.10 and 6.15	247
12.14.1. Install and configure interface to Btrieve	248
12.14.1.1. Advanced options.....	249
12.14.1.1.1. Open mode	250
12.14.1.1.2. Name suffix	251
12.14.1.1.3. Name options	252
12.14.1.1.4. Parameters	253
12.15. ODBC driver setup.....	254
12.16. ODBC drivers interface - Application type	255
12.16.1. Type 1 - Support of Informix Database table locks	256
12.16.2. Type 4 - Always perform ORDER BY	257
12.16.3. Type 4096 - Alpha fields may NOT be NULL	258
12.16.4. Type 8192 - ODBC Drivers with support of one connection per table only	259
12.17. ISAM database interface - Application type	260
12.17.1. Type 2 - Support of SAMSON system	261
13. Other drivers / Notes on some application packages.....	262
13.1. GLOBAL 3000 Speedbase and Global 2000 ISAM/DMAM	263
13.2. ALX database	264
13.3. SSV database interface	265
13.3.1. Write sortindex	266
13.3.2. Headerlength	267
13.3.3. First data record.....	268
13.3.4. Make indexfile > records	269
13.3.5. Write CRLF	270
13.3.6. Indexfile naming	271
13.3.7. Duplicates >= index	272
13.3.8. Read CRLF.....	273
13.3.9. Compression used	274
13.3.10. Initial codetable.....	275
13.4. Extended SSV interface.....	276
13.4.1. Fixed recordlength.....	278
13.4.2. Fixed blocklength	279
13.4.2.1. Max no. of data records.....	280
13.4.3. Fixed records/block	281
13.4.4. First free record	282

Data-Dictionary

13.4.5. No.of free records.....	283
13.4.6. Next free record	284
13.4.7. Filename type	285
13.4.8. Field Pack types	286
13.4.9. Key Pack types.....	287
Figure list	288
Index	290

1. Preface

The Data-Dictionary is used to setup the database. A database is a collection of tables containing values allowed you can receive.

Each table refers to a table type. The type, may for example, be ODBC, C-ISAM or Btrieve.

All tables hold a description of elements, also known as fields, each described with a number, name and format. Unlike from other database systems the Data-Dictionary allows use of blanks and special characters in the field names, and definition of a logical format, later used in SW-Tools TRIO, e.g. output format for reports/enquires and input format for data entry applications.

A table also contains index definitions, defining how the data is ordered in the physical file, e.g. an article file normally has a primary index defined as article number and might have a secondary index of supplier. With the indexes you will get fast access to any value in the table.

An important feature in the Data-Dictionary is the facility to define the relations between tables once and for all. When the definitions have been made, features in SW-Tools TRIO provide automatic graphical relation diagrams, relations to fields from other table fields in reports and enquiries, and runtime interaction between multiple running enquiry applications are available.

All in all the Data-Dictionary provides a complete database description system with the following feature:

- **Definition of tables**
- **Definition of elements and the formats**
- **Definition of indexes**
- **Definition of relations**
- **Documentation on tables and fields**

1.1. Licence information

The Data-Dictionary is copyrighted by SW-Tools and your licence information is shown briefly whenever the program is started.

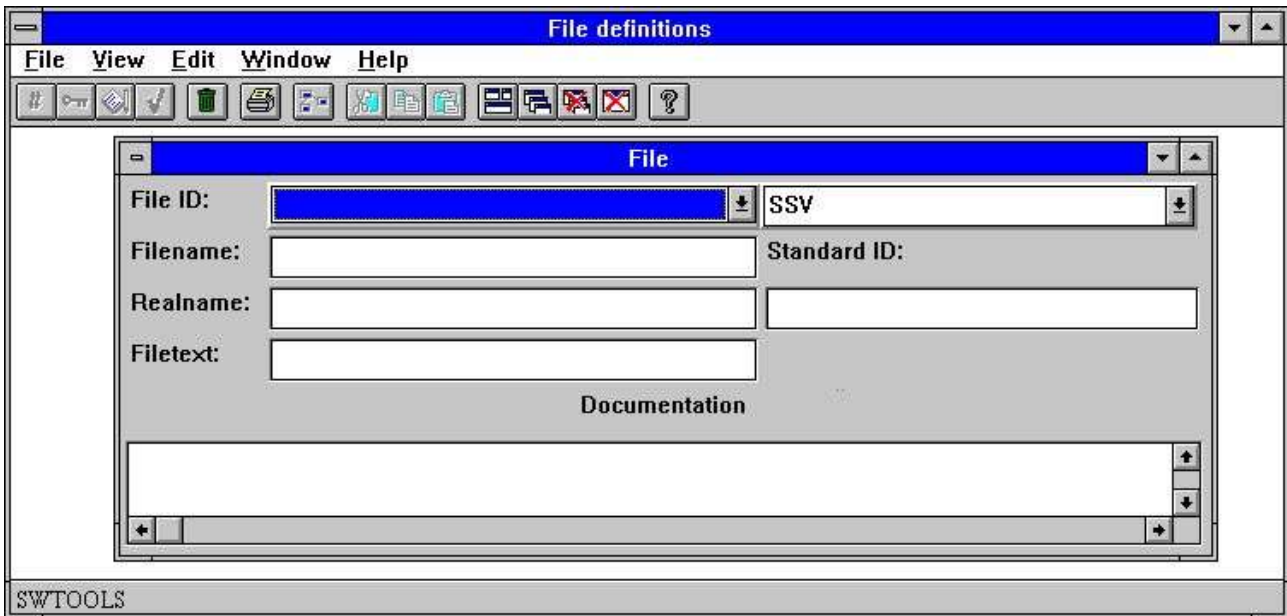


1. Licence screen

You are of course only allowed to use the programs according to your licence agreement.

1.2. Start of the Data-Dictionary program

When the Data-Dictionary is started the following is displayed:

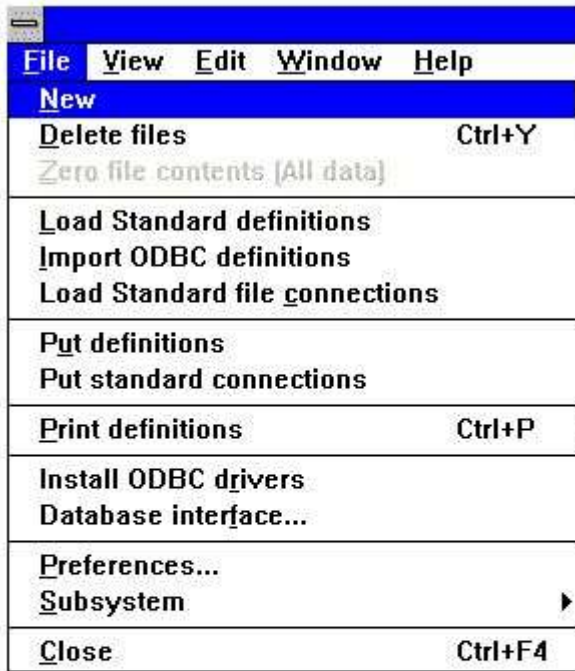


2. Data-Dictionary

1.3. About the user interface

To access the functions in the Data-Dictionary you may use the menus or the related buttons on the toolbar.

• The pulldown menu



To print the table definitions you may choose the function from the menu or by clicking on this button on the toolbar.

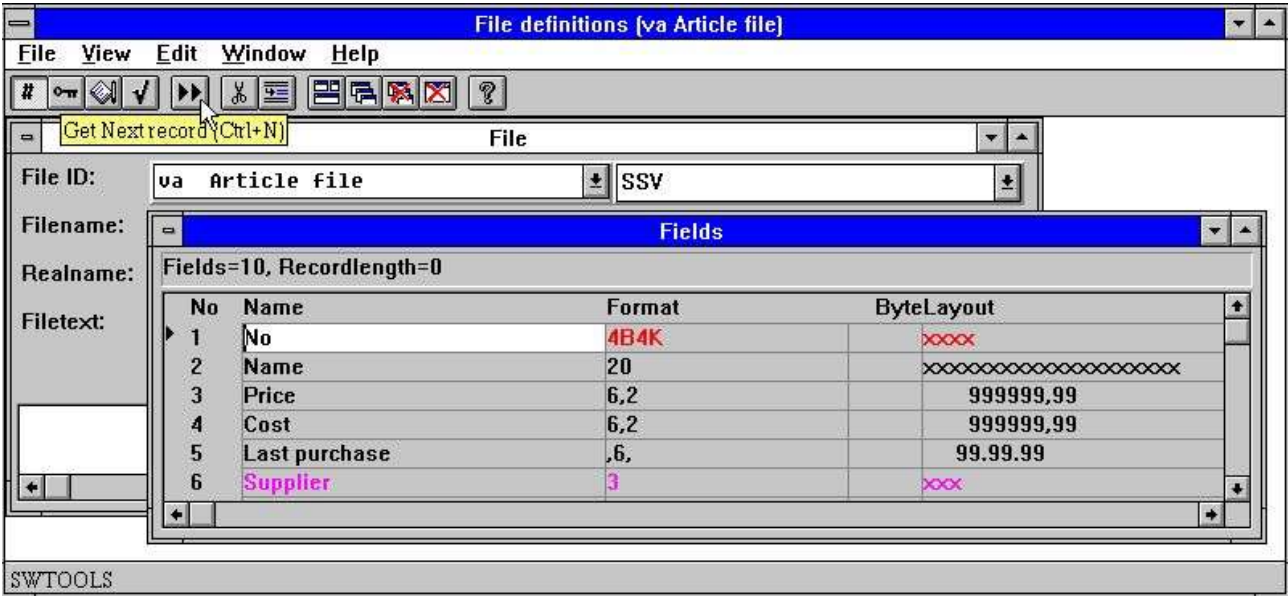
• The toolbar



There is a short description attached to each function button on the toolbar. It will be displayed whenever the mouse-cursor is over the button area.

3. Accessing the functions using menus or the toolbar

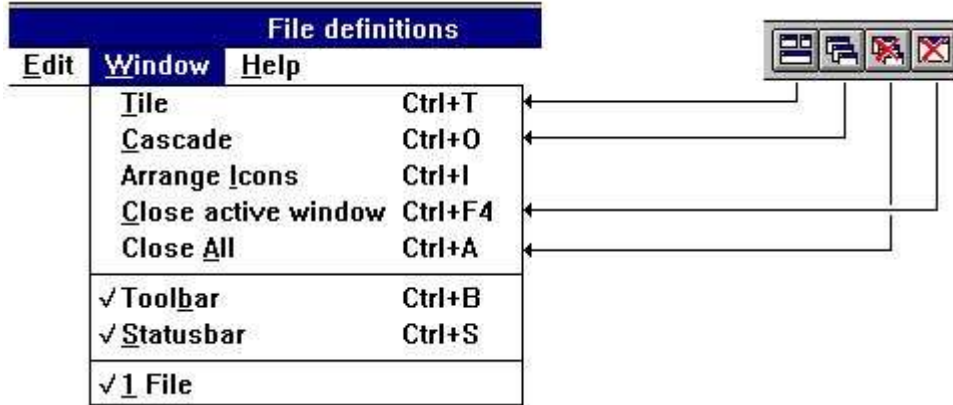
Because the toolbar does not have room for all the functions in the Data-Dictionary it will vary from function to function, as for the pulldown menus. For example you will see this toolbar when you activate the field function:



4. Other buttons on the toolbar

While the menus and toolbar buttons change between functions the Data-Dictionary has some general functions which always offers, to position windows, close the active window and to access the on-line manual.

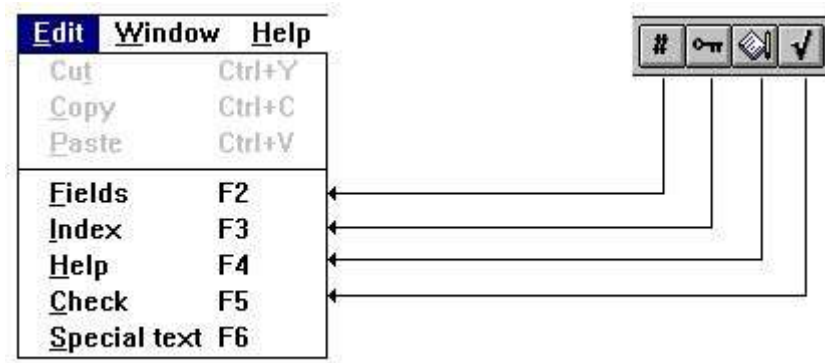
Because these functions are on the toolbar they can also be accessed from the pulldown menus:



5. General menus and buttons

1.3.1. Functions for table definition and amendment

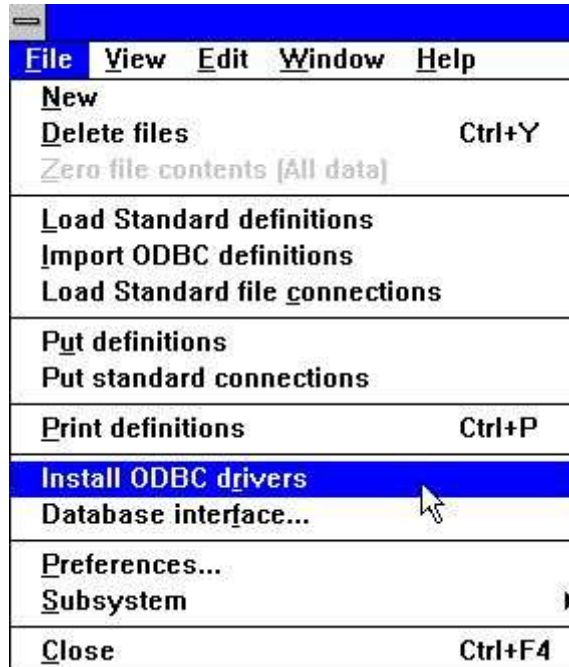
The Data-Dictionary has the following functions to process tables.



6. Functions

2. Driver installation

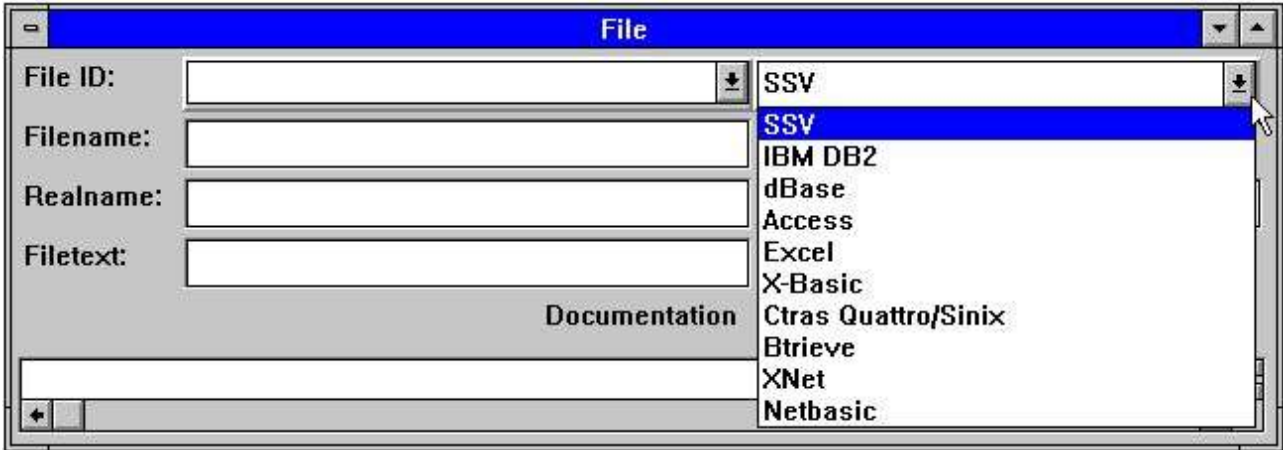
The driver installation function is selected from the following menu:



7. Selecting the driver installation function

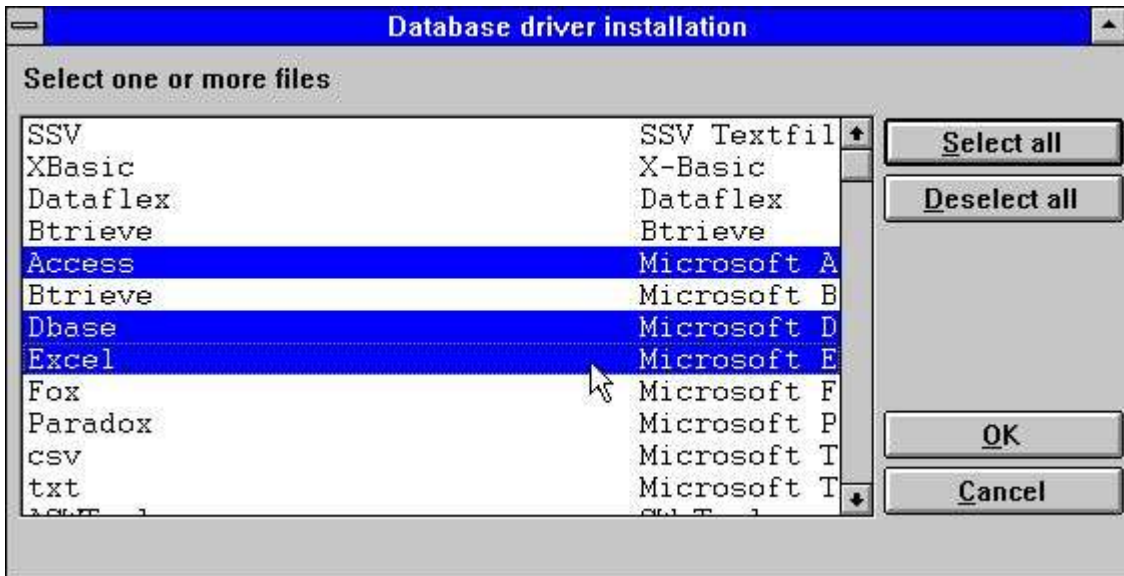
2.1. Installation of drivers

Each table defined in the Data-Dictionary has to be linked to a Driver. E.g. the demo files are linked to the internal SW-Tools SSV Textfile driver, a simple semicolon separated textfile. To check if a driver is installed, view the current list:



8. Installed drivers

Other drivers may be installed selecting the following function:

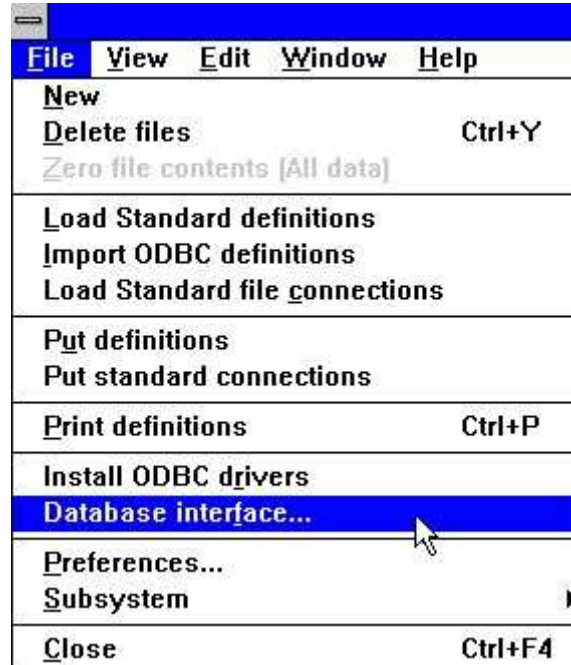


9. Installing new drivers

This function allows selection of one or more drivers. If the selected driver is already installed it is replaced by the new one, provided that the driver name remains the same. If it is changed the system may use the same driver with different options.

3. Database interface

The driver interface function is selected from the following menu:



10. Selecting the driver interface function

3.1. Database interface parameters

Installed drivers are controlled by a set of parameters. Present for all drivers are standard path, server name and password, how characters are to be converted, e.g. codetable from UNIX to DOS. Drivers may provide some extra parameters and that can be changed using the advanced option.

To change standard driver parameters select the following function:

Database interface	
Type	Access
Internal Type	Open Database Connectivity
Description	Access
Database	dbq=c:\odbc\sd\k\smpl\data\access\sam
User	
Password	
Data Source (Driver)	Access
Connection type	Normal
Server password	
Code conversion	013 ODBC OemToAnsi
Read only	No.

11. Changing driver parameters

3.1.1. Type

The file type is the actual installed driver. By selecting a driver the dialogue allows changes to the parameters for this driver.

3.1.2. Internal type

The internal file type selects the physical driver type. This value is held internally and can be any of the installed drivers.

3.1.3. Description

The description is used whenever a list of installed drivers is shown. Therefore it is recommended you enter a meaningful name here.

3.1.4. Database

A database is a collection of one or more tables, located somewhere on the system. For ODBC you can state driver options as `DBQ=c:\Access,OPT=Y`

3.1.5. User name and password for ODBC databases

The user name and password is provided for the ODBC-interface. All drivers with the internal type set to ODBC may need a user name and password to logon to the database. Refer to the system administrator in order to get this information.

3.1.6. Data source, Connection type and Server password

The data source (server name) and password is provided for ODBC and Windows Sockets TCP/IP drivers. The server name refers to a host name for a UNIX machine when the connection type is Windows Sockets TCP/IP and the data source name when ODBC.

The connection type has to be set to Normal for ODBC drivers. All other drivers may be set for Windows Sockets TCP/IP if the database is located on a UNIX machine.

The Windows Sockets TCP/IP connection only requires a server name. This name can be entered as the IP-address or, if defined in the HOSTS file, a server name. For example an IP-address might be

200.0.0.1

If the connection type is Windows Sockets TCP/IP the SW-Tools UNIX server must be installed. (See later).

3.1.7. Code conversion - Language dependent and special characters

All drivers can use a code conversion table when reading data from a database.

If a driver is connected with Windows Sockets TCP/IP and the database is located on a UNIX machine, language dependent and special characters may not be retrieved correctly without code conversion.

The Data-Dictionary supports a number of code conversion tables. The following gives an example of the possible code tables:

0	None
1	UNIX/DOS Danish
2	UNIX/DOS German
10	ISO8850
13	ODBC (OemToAnsi)

3.1.8. Read only

Some of the drivers may not support updates of the databases. Please refer to the documentation for the driver itself to check if updates are possible.

3.2. Advanced driver options

A driver may require some extra options specific to this interface such as Table owner for ODBC drivers and Company number for Basic systems.

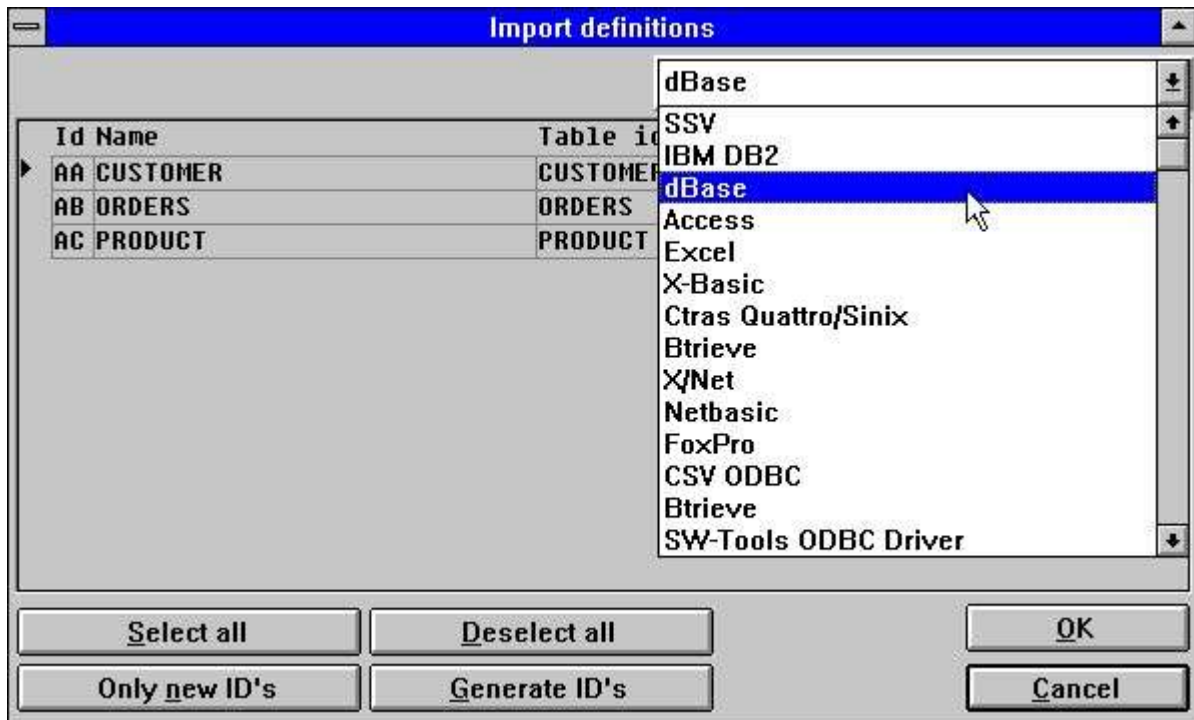
3.3. Deleting an installed driver

If an installed driver is not needed it can be deleted by selecting the delete button. This function does not delete the file definitions that are attached to the driver. This has to be done manually.

4. Importing ODBC definitions

4.1. Importing ODBC table definitions into the Data-Dictionary

When an ODBC driver has been installed it is possible to import table definitions directly into the Data-Dictionary, e.g. tables defined in Access or Excel may be imported for use in SW-Tools TRIO.



12. Importing ODBC definitions

4.1.1. Driver type

Before importing any table definitions a driver must be selected. When the driver is selected the screen shows all tables accessible, if any.

4.1.2. Id

Each table is uniquely identified by an ID of two characters which is generated automatically as AA,AB,AC...

You may choose another ID yourself or blank out an ID to skip a single table. NOTE that existing definitions with same ID will be overwritten !

By using the button 'Only new IDs' all suggested IDs already in use can be blanked out.

By use of the button 'Generate IDs' new IDs can be generated for all tables as A0,A1,A2...

4.1.3. Table name

The table name is a free text taken from the driver. You may amend this name before importing the definition. The name is displayed together with the ID when you later access the table.

4.1.4. Table-id

The table-id is the physical filename. For ODBC tables it is the physical table name equal to the one entered when the table was created. This cannot be changed here but may be changed after importing the definitions.

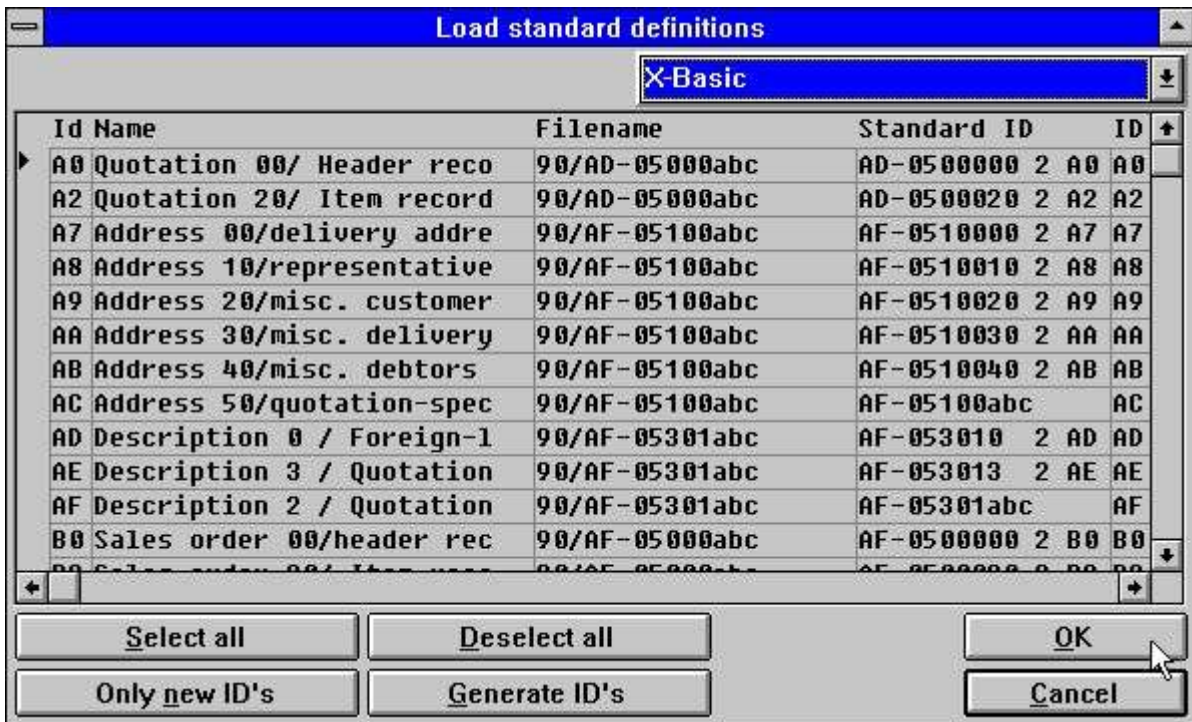
4.1.5. Suggested id

This column displays the generated id. If the id in the first column has been changed this is ignored.

5. Standard definitions

5.1. Loading standard table definitions from a textfile

The Data-Dictionary allows loading of standard table definitions from a text file. If you already have the BASIC version of SW-Tools File definitions installed, it is possible in the file definition module to enter the PUT command and hereby produce a textfile. This textfile can then be loaded using the following function:



13. Standard definitions from the BASIC version of SW-Tools File definitions

5.1.1. Driver type

Before loading the definitions you should select a driver type. The definitions will then be given this file type during the load.

5.1.2. Id

The tables from the textfile is displayed with the ID from the origin system and may be amended before loading.

You may choose another ID yourself or blank out an ID to skip a single table. NOTE that existing definitions with same ID will be overwritten !

By use of the button 'Only new IDs' all suggested IDs already in use are blanked out.

By use of the button 'Generate IDs' new IDs are generated for all tables as A0,A1,A2...

5.1.3. Table name

The table name is a free text taken from the driver. You may amend this name before loading. The name is displayed together with the ID when you later access the table.

5.1.4. Filename

This is the physical filename and may be amended before load. This may be used for changing the disknumber for a group of Basic files.

5.1.5. Suggested id

This column displays the original id comparable to the first column ID which may have been changed.

5.2. Loading standard file connections

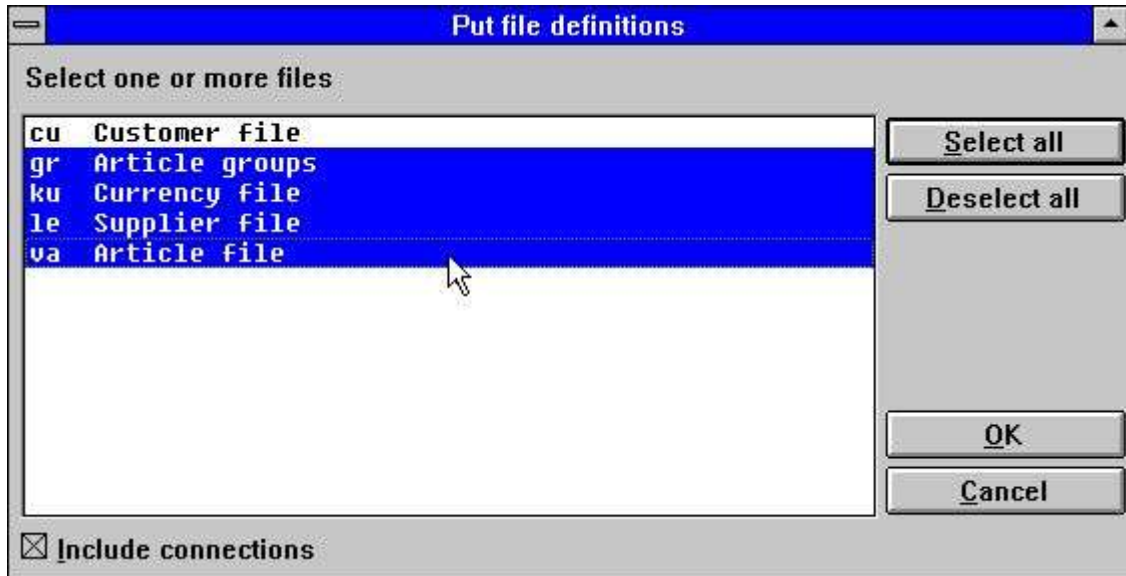
When a set of master file definitions are distributed the file definitions are saved using the normal PUT of file definitions, but the file connections are kept as a separate file. Each file is marked with the original systems ID and a STANDARD id (SID) which identifies the definition even when the ID is changed.

For a LOAD of standard file connections the system first looks for which files are activated based not on the file ID but the originating SID identification. Only connections between active files are loaded and during the load the IDs are modified according to the active system.

Standard file connections are marked first in the name with ==> and on the LOAD old connections will be removed and replaced. Please keep this in mind if you modify standard connections: If you do not remove the ==> mark, your modifications are lost if you load the standard connections again.

5.3. Put of table definitions to a textfile

A number of file definitions may be stored in one textfile with the PUT function and later loaded on another system or another computer using the GET function.



14. Putting definitions to a textfile

5.4. Put of file connections to a textfile.

This function produces a textfile with the connections between the selected files for later loading using the LOAD standard file connections. (See above.)

The SID may be AF-0500002031AS for the COMET file AF-05000 record type 020 Comet version 3.1 stored on the master with file id AS.

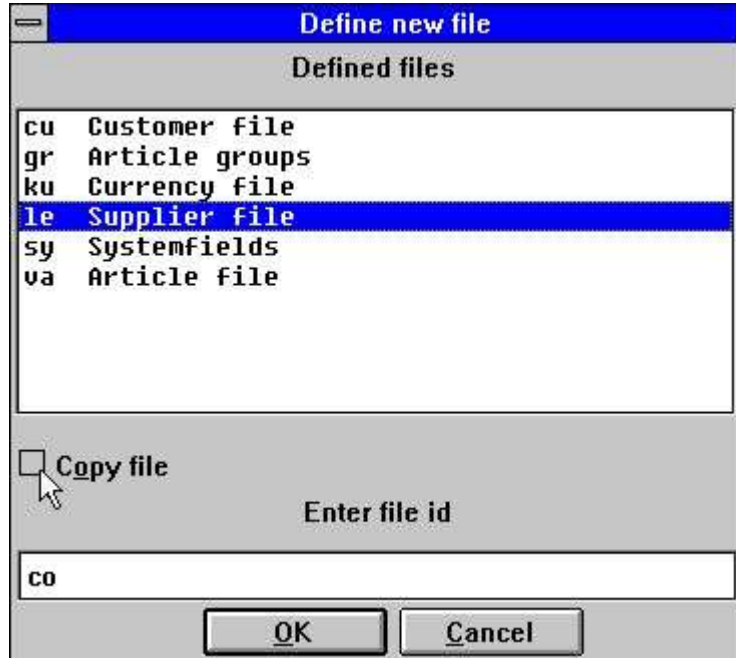
By load/put of connections only the first 13 characters of the SID is used, the file ID is set automatically. The SID will default to the file ID if not entered.

6. Table information

A table is a description of how information is stored. It define the type of database used, the table name or physical path of the file.

6.1. Creating a new table

When creating a new table a unique id is required. The rules for the id are described later in this chapter.



15. Id of new table

If the new table is to be based on a previously defined table please select the table from the list of defined tables and checkmark the 'Copy file' option. The Data Dictionary will copy all definitions of fields, index, help etc. from the selected table to the new.

6.2. Table parameters

6.2.1. Id

The table id must be entered as two characters. The first character must be a letter the second can be a letter or a digit, e.g. X1.

The following id are reserved for system use:

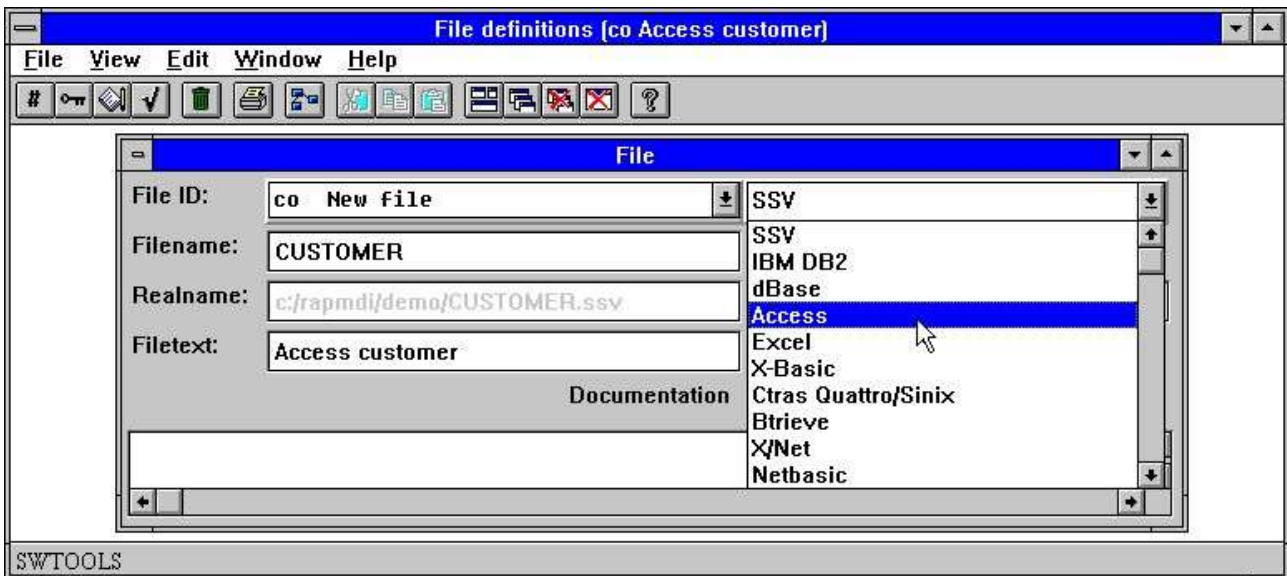
SY **System fields**
WW **Work fields**

6.2.2. Type

The table types allowed depends on which database drivers are installed. For example a system may include the following types:

- **ODBC drivers (Access,Excel etc.)**
- **C-ISAM**
- **Btrieve**
- **X-BASIC**
- **UNIBASIC/SURFBASIC**
- **others**

If the required type id not found in the list please refer to the chapter 'Driver installation'.
If a table it to use the Access ODBC driver select the following:



16. Setting the driver for a table

6.2.3. Name

The table name is the physical name. For a file type such as Btrieve it may be entered as a physical path as

C:/BTRIEVE/CUSTOMER.DAT

or if it is an ODBC table of some kind just as it is: CUSTOMER

6.2.4. Real name

Based the selected table type and the entered table name a real name for the table is retrieved. The displayed name must match the actual disk file of the table on the actual server, otherwise reading from the table is not possible.

The real name is generated using information from the driver attached and the name for the table. If the driver is X-Basic and the standard path by installation is set to

/X.BASIC/0/

and the name of the table is

90/LF-06000abc

the real name may be generated as

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

6.2.5. Text

The table text is a logical name for the table. The name may contain letters, digits, blanks and special characters. The text is displayed together with the ID whenever you use the table.

6.2.6. Documentation

The table can be described here in free text. The description is printed when printing the table documentation.

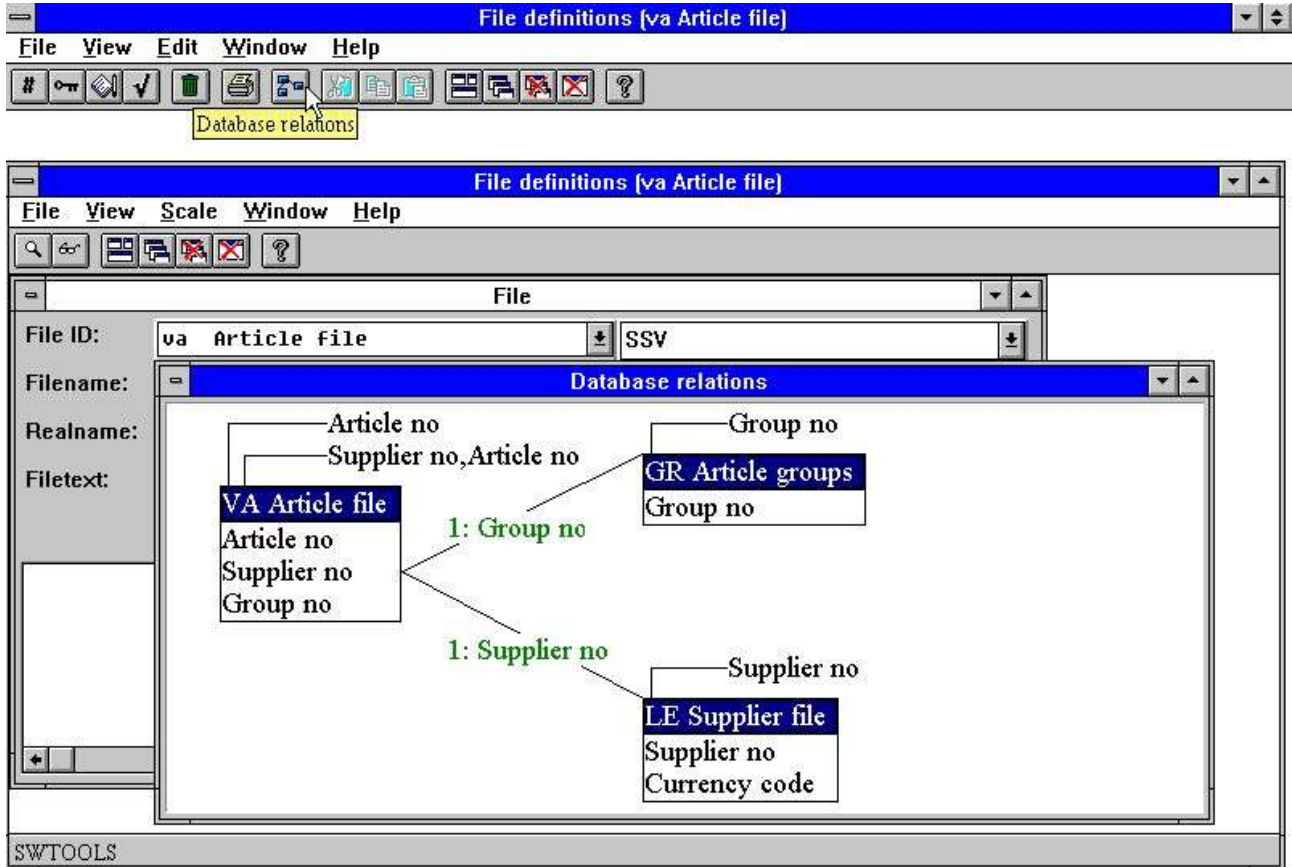
6.2.7. Standard id

The standard id identifies the table when originating from an appropriate software package. The SID is used by LOAD and PUT of standard table connections, see above, you are not required to fill the SID if you do not intend to use these functions.

The SID may be AF-0500002031AS for the COMET file AF-05000 record type 020 Comet version 3.1 stored on the master with file id AS.

6.3. Database relations enquiry

Each table may have relationships to other tables defined. When relations exist between tables these can be displayed graphically or printed. To view relations from one table to others select:

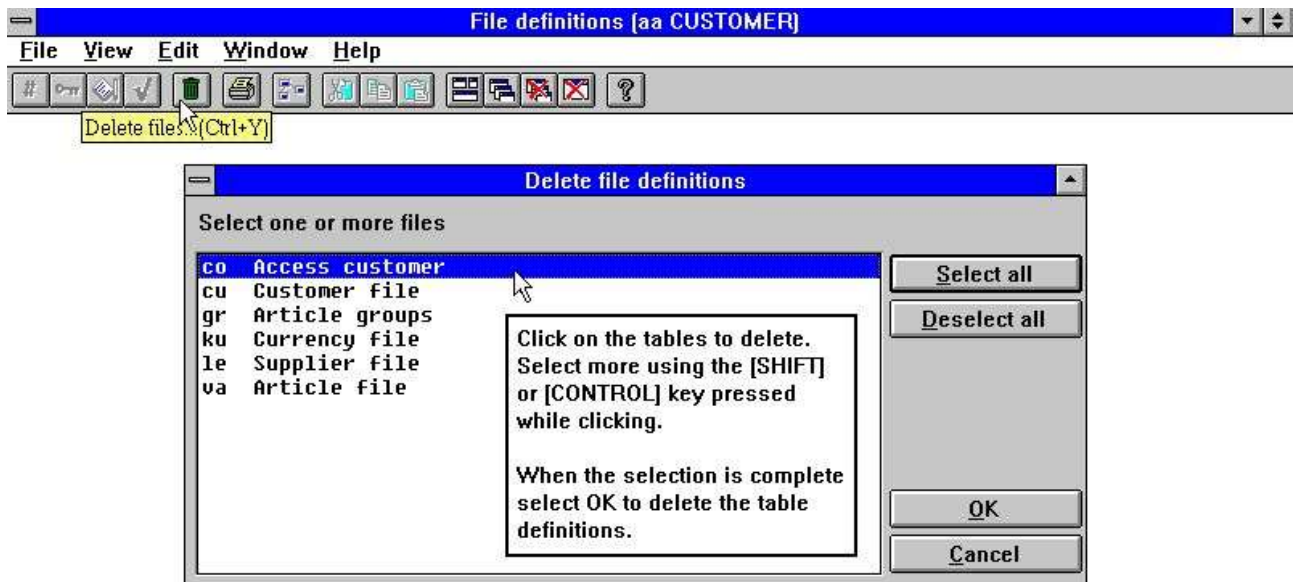


17. Database relations enquiry

The display can be scaled, and can generate the diagram in one or more levels.

6.4. Deleting tables

Deleting defined table is done by selecting the following function:



18. Deleting tables

6.5. Record overview

The record overview displays an overview of the first 100 records in a table. When displaying records from a BASIC file the control record will be show in the FDF module as data from this may be defined.

Record overview				
Refresh		1-99		
No	CustID	Company	Address	City
1	ANDRC	Andre's Continental Food Market	P.O. Box 209	Bell
2	ANTHB	Anthony's Beer and Ale	33 Neptune Circle	Clift
3	BABUJ	Babu Ji's Exports	Box 29938	Lond
4	BERGS	Bergstad's Scandinavian Grocery	41 S. Marlon St.	Seat
5	BLUEL	Blue Lake Deli & Grocery	210 Main St.	Port
6	BLUMG	Blum's Goods	143 Blum Rd.	Lond
7	BOTTM	Bottom-Dollar Markets	23 Tsawassen Blvd.	Tsav
8	BSBEV	B's Beverages	Fauntleroy Circus	Lond
9	CONSH	Consolidated Holdings	12 Berkeley Gardens	Lond
10	EASTC	Eastern Connection	35 King George	Lond

19. Record overview

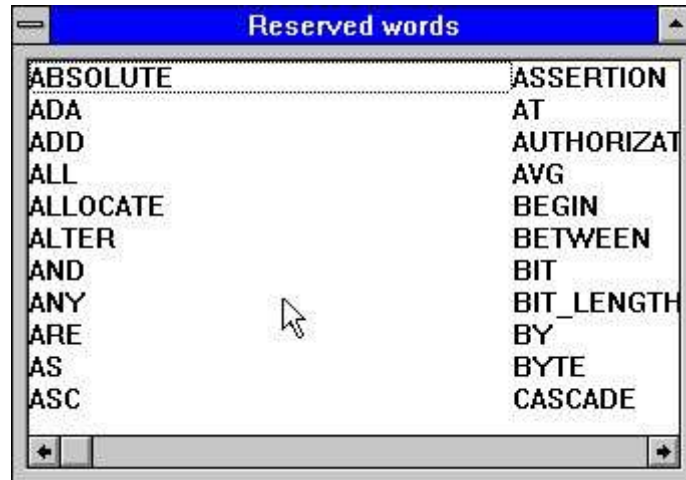
If the record overview comes up empty the following might be wrong:

- **The driver type**
- **The table name**
- **No records in the table**

A subwindow is shown which will contain eventual error messages during open or read.

6.6. Reserved words

The reserved words function displays an overview of reserved words in ODBC. This is useful when entering table name etc. for a driver is not allowed to use one of the reserved words.



20. Reserved words

7.1. Number

The field number cannot be entered but is attached to each field defined. A field is always selected by number in SW-Tools TRIO. It is however, also possible to refer to a field using the field name, but not recommended.

7.2. Name

The name may contain all letters, digits, blanks and special characters. It is normally used as standard heading on reports or enquiry applications.

7.3. Format

The field format is a logical format. It defines how the values are retrieved and stored in the table. It also defines how the field has to be printed or displayed. The format can define:

- **An alphanumeric field**
- **A numeric field**
- **A date field**

An alphanumeric field may contain text of any kind and is only limited by a maximum length. Therefore the field can be defined as:

- 10 The field is limited to 10 characters
- 20 The field is limited to 20 characters

A numeric field contains only numeric values but may be defined with or without a decimal point. The following example describes the syntax:

- 2, The field may contain the value from 0 to 99
- 2, The field may contain the value from -99 to 99
- 5, The field may contain the value from 0 to 99999
- 7,2 The field may be negative and have a maximum of 7 digits and 2 decimals
- 9,3 The field have a maximum of 9 digits and 3 decimals

A date field may be one of the following:

- ,6, Contains the date as YYMMDD, where YY=Year, MM=Month, DD=Day
- ,8, Contains the date as YYYYMMDD, where YYYY=Year, MM=Month, DD=Day

7.3.1. Editing of numeric fields

A numeric field value can be edited in different ways. Values can be displayed in comma separated thousands just by inserting a leading comma in the format

,9,3
-123,456,789.123

A field defined with leading comma will always support negative values. The edit character (, or .) may be set in the preferences menu.

Please note that date fields with the format ,6, and ,8, must be given as ,6,0 and ,8,0

Leading zeros may be defined by placing an & in the format:

9,3&
000012345.123

Leading * for amount fields may be defined by placing a * in the format:

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

The sign may be defined leading or trailing

9,2-
12345.12-

Zero suppress (blank if zero) is defined by a ' in the format as 9,2'

7.3.2. Floating decimals

By use of 9,5? you may now define a field which is output with 0 to 5 decimals as 123 or 123.45678 dependent on the actual field value.

Together with the field format a number of extended options may be used to define the storage in the record. As the Data-Dictionary supports several different database systems and file types, also many different field types are available.

7.3.3. (nnn) Table fields

2(010)

defines a field as 2 alphanumeric characters which occurs 11 times numbered from 0 to 10. The field value follows consecutively in the record. Such fields can be used in calculations as #7(0),#7(1),...,#7(11)

7.3.3.1. (nnn+) Grouped table fields

2(010+)

4,(010+)

8,(010)

defines a group of table fields occurring 11 times. The fields are stored in the record as 2 characters followed by 4 digit numeric and 8 digit numeric which is then repeated 10 times.

7.3.4. W Work fields

10W

defines a field as a work field that is a field not taken from the file record by reading. Work fields are used for storing the key value in files where the key is not contained in the record itself.

7.3.5. (nn) Explicit record length

4(16)

defines an alphanumeric field of 4 characters but stored in the record as 16 characters. This means there will be 12 bytes reserved for some other use, typically for extension of keyfield length without changing the complete record layout.

7.3.5.1. (nn,x) Explicit record length in bits

4(16,4)

defines a field occupying 16 bytes and 4 bits (16½ bytes) in the record.

7.3.6. Bnn Explicit byte number

8,B100

defines a field starting on byte number 100 in the record. The bytes are numbered from 0 and onwards and you should note that some database systems align fields on specific boundaries, e.g. in Basic all numeric fields are on word (16 bit) boundaries.

7.3.6.1. Bnnn,x Explicit bitnumber

8,B45,4

defines a field starting on byte number 45+4 bits (byte 45½)

7.3.7. Tn Fixing field type

4,T1

defines a field with 4 digits of type 1 in the given database system. An example is on Basic systems where 4 digits are forced into a 1 word variable which has a limitation of 7999.

7.3.8. Unn Specific codetable for single fields

32U2

A codetable number can be given individually for each field overriding the codetable for the file / database system.

7.3.9. R Right justified display

10R

Causes an alpha to be displayed right justified. Note also index keys is formed with the field right justified.

7.3.10. S Stop (delimiter) character in alphanumeric field

12S

Defines a split of read statements in a Basic program after this field. Valid but has no importance for any of the TRIO C programs.

7.3.11. Fnn System field addressing

This option occurs in the SY file only and is used to bind fields as #RECNO to the specific file.

7.3.12. K,D - DATAMASTER Index fields

4K

defines an unique key in TRIO DATAMASTER files, ignored for other files.

4D

defines a key with duplicates allowed in DATAMASTER files, ignored for other files.

7.3.12.1. E,I,V,X,Z - DATAMASTER Special index fields

These options are retained for compatibility with the Basic DATAMASTER system but not used.

7.3.13. Cnn - DATAMASTER check for field

This option is reserved for compatibility with the Basic DATAMASTER system but not used.

7.3.14. Qnnn Explicit SQL type

19Q11

defines a **TIMESTAMP** field for use by ODBC access of this field. The SQL types are normally set by importing definitions, but especially:

- 9 DATE**
- 10 TIME**
- 11 TIMESTAMP**

7.3.15. An Access mode

A0 enables read and write, A1 read only.

This option is used for ODBC SQL statements where A1 should be used to keep field as automatic counts away from any UPDATE statement.

7.3.16. Pnn Packed fields

Packing of fields determines how the field is stored in the database record and is highly dependent upon the database system the file is using.

8,P

defines a packed numeric field with 8 digits, but how this field is packed depends on the database driver.

4P2

defines an alphanumeric field of packtype 2 dependent of the database driver. You should refer to the documentation for your individual database system to find the pack code supported.

7.3.16.1. Pnnnn **Explicit specification of the pack type**

The packtypes 0-999 are reserved for individual database drivers use whereas 1000-9999 are packtypes which are the same for all drivers.

,6,P1040

will always define a ODBC DATE structure even if this field happens to be stored into a Basic file.

Type	Name	Is stored as	Usage
0-999		Driver dependent	
1000	LDCHAR	Alphanumeric with trailing blanks	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-variable short	C
1006	LONG	C-variable long	C
1007	FLOAT	C-variable float	C
1008	DOUBLE	C-variable double	C
1009	FLEXBCD	BCD	Dataflex
1010	FLEXBCDDATE	BCD-date	Dataflex
1011	CHAR	C-variable char	C
1012	UCHAR	C-variable unsigned char	C
1013	SCHAR	C-variable signed char	C
1014	NUMCHAR	Not implemented	C
1015	ULONG	C-variable unsigned long	C
1016	UINT	C-variable unsigned short	C
1020	BTINT1	Integer 1-byte	Btrieve
1021	BTFLOAT	Float 4-bytes	Btrieve
1022	BTDATE	Date	Btrieve
1023	BTTIME	Time	Btrieve
1024	BTDECIMAL	Decimal	Btrieve
1025	BTLOGICAL	Not implemented	Btrieve
1027	BTNUMERIC	Numeric	Btrieve
1028	BTBFLOAT	BFloat	Btrieve
1029	BTLSTRING	LString	Btrieve
1030	BTZSTRING	ZString	Btrieve
1031	BTUBINARY	Not implemented	Btrieve
1032	BTAUTOINC	Not implemented	Btrieve
1033	BTINT2	Integer 2-byte	Btrieve
1034	BTINT4	Integer 4-byte	Btrieve
1035	CONNUM	Varlength double amount	Concorde
1036	CONNUM4	Varlength long	Concorde
1037	CONALF	Varlength string	Concorde
1038	CONDATE	Date	Concorde
1039	CONNUM2	Unsigned int	Concorde
1040	ODBCDATE	Date structure	ODBC
1041	ODBCTIME	Time structure	ODBC
1042	ODBCSTAMP	Timestamp structure	ODBC
1043	RCHAR	Right justified string	Concorde
1044	CONDBL	Fixedlength double amount	Concorde 4
1045	CONDAF	Fixedlength date	Concorde 4

Data-Dictionary

1046	RMIDBL	RM double	C-RM
1047	STRDATE	Date YYYYMMDD as string	C-RM
1048	ODBCSL1	String minimum 1 char	Access
1049	LDINTI	As LDINT, but bytes other way round	CX-Basic
1050	LDLONGI	As LDLONG, but bytes other way round	CX-Basic
1100	PARITY	Alphanumeric with parity	Basic \$
1101	BCD1	1-word BCD integer	Basic 1%
1102	BCD2	2-word floating point number	Basic 2%
1103	BCD3	3-word floating point number	Basic 3%
1104	BCD4	4-word floating point number	Basic 4%
1105	BCD5	5-word floating point number	Unibasic 5%
1106	CALL60	Value packed with CALL 60	Basic packed
1107	NX2	Value packed with CALL 60, no decimals	Basic
1108	NX3	Field stored alphanumeric with decimals	Basic
1109	NX4	Field stored alphanumeric without decimals	Basic
1110	NX5	Stored as 1%, no decimals	Basic
1111	BINARY	Binary stored	Cobol comp-3
1112	GSM1	Speedbase alpha (x bytes)	Global
1113	GSM2	Speedbase 1-2 digits numeric (1 byte)	Global
1114	GSM3	Speedbase 3-4 digits numeric (2 bytes)	Global
1115	GSM4	Speedbase 5-6 digits numeric (3 bytes)	Global
1116	GSM5	Speedbase 7-9 digits numeric (4 bytes)	Global
1117	GSM6	Speedbase >9 digits or (dig+dec)/2 bytes	Global
1118	GSM7	Speedbase date fields uses fna()/fnb()	Global
1201-19	CROSS	BCD: S exp bcd 01-19 digits	C-Isam
1221-39	CROSS	BCD: bcd 01-19 digits exp S	C-Isam
1241-59	CROSS	BCD: bcd 01-19 digits	C-Isam
1053	LDCHARR	Right justified field no extra byte as 1043	C-Isam
1121	NAVI_STR	Textstring	Navision P5
1122	NAVI_DATE	Date structure	Navision P6
1123	NAVI_TIME	Time structure	Navision P3
1124	NAVI_BLOB	Binary large object	Navision
1125	NAVI_BOOL	Boolean	Navision P2
1126	NAVI_S16	Short	Navision P7
1127	NAVI_S32	Long	Navision P8
1128	NAVI_U8	Char (byte)	Navision P1
1129	NAVI_ALFA	Textstring with 1.byte=Length	Navision P
1130	NAVI_BCD	Amount	Navision P9
1131	NAVI_O32	Optioncode	Navision P4

7.3.17. Vxxx Secondary packtypes

Due to year 2000 updates it has been necessary to introduce a secondary packtype which can be seen as a fixed calculation performed just after the field has been read and unpacked according to the Pxxxx packtype, or just before the field will be packed and written to the record.

The secondary packtype is given as Vxx within the field format as: ,6,V1

Type	Name	Calculates	(Default) Parameter ZZ
V	YYMMDD	Shortdate YYMMDD -> YYYYMMDD	(50) Year 2000 when YY<=ZZ
V1	DDMMYY	Shortdate DDMMYY -> YYYYMMDD	(50) Year 2000 when YY<=ZZ
V2	DDDDDD	Daynumber DDDDDD -> YYYYMMDD	(0) Baseday V2.BASENUMBER
V3	NXU2000	Basic unpacked ZYmmdd, Z=?@ABC --> 00-09	None
V4	NXP2000	Basic packed ZYmmdd, Z=space+,-. --> 0-4	None
V8	OCTAL	Convert from octal string by read, "1234" -> 668	None
V16	HEXA	Convert from hex string by read, "abcd" -> 43981	None

7.3.17.1. Secondary packtype parameters

A parameter may be given together with the secondary packtypes as: ,6,V1.30

This may be used to overrule the default year wrapping by converting to/from short date formats.

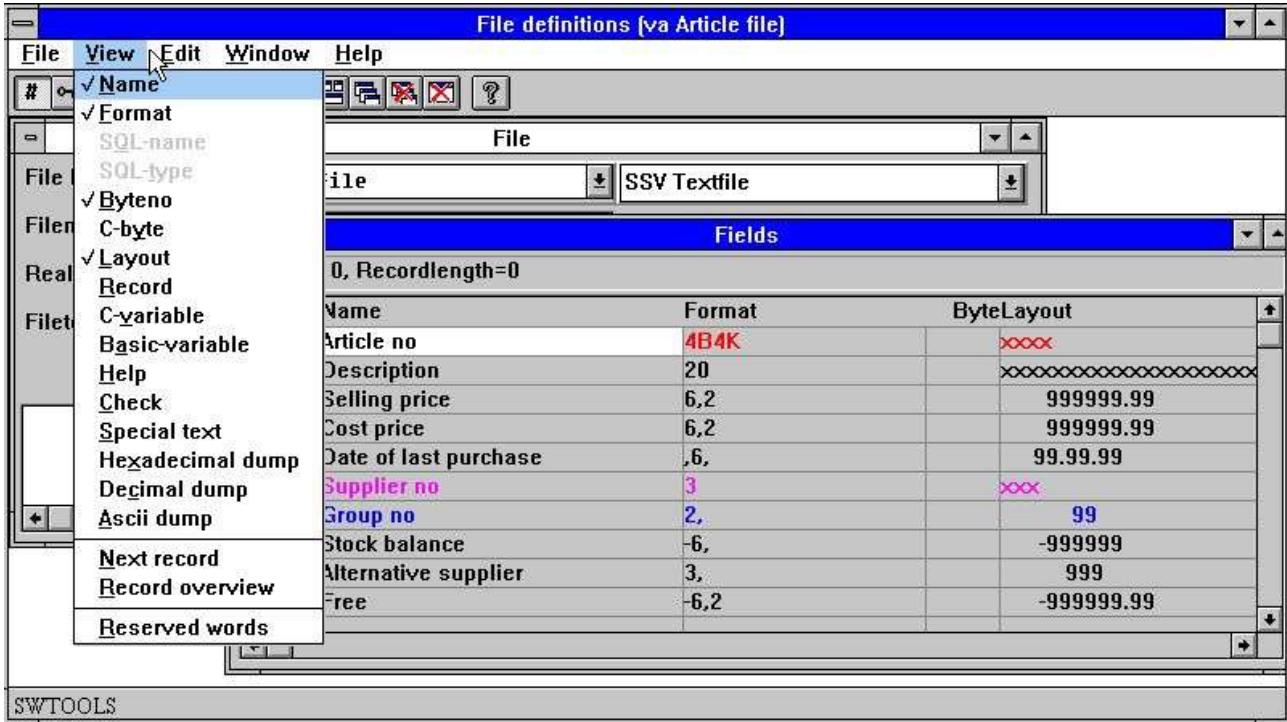
7.3.17.2. Default secondary packing for BASIC dates

For short date fields (,6,) in BASIC files the secondary packtype will default to V for numeric fields, V3 for alphafields and V4 for packed fields.

These rules are also applied to any keyfields containing short dates.

7.4. Information on the field line, the VIEW menu

By default the byte number and the layout of the field will be displayed on the field line. By use of the VIEW menu you may change this to display other columns for each field:



23. The VIEW menu

7.4.1. Byte number

The byte number is automatically calculated and displayed whenever needed. You may use the Bnnn option in the field format if you need to state a specific byte, e.g. if not all fields are defined or the sequence does not follow the field sequence in the record.

7.4.2. Field Layout

The layout displays how the field is printed or displayed. If the field is alphanumeric a number of X's corresponding to the maximum length of the field is displayed. For numeric fields 9's are shown.

7.4.3. SQL-names

The SQL names are only accessible when the table type is ODBC. The SQL-name enables you to have field names usable for headers different from SQL_Names used by the database driver. The SQL-Name is passed directly to the ODBC driver if present which means these must follow the syntax supported by the driver. The names should not contain any blanks or special characters unless enclosed in '...'.
 In the EDIT menu you have the option to generate the SQL-Names from the field names following standard rules. However you may also enter these yourself which allows use of special columns as COUNT(*) and A+B

No	Name	Format	SQLname	SQLtype
1	Article number	-5,	ProdID	LONG
2	Name	40	ProdName	TEXT[x]
3	Price	-18,4	UnitPrice	CURRENCY
4	My own text	64	Descript	TEXT[x]
5	Weight	-15,	ShipWt	CURRENCY

24. SQL-Names and SQL-Types

7.4.4. SQL-types

The SQL types are only accessible when the table type is ODBC. The type names displayed are the default according to the field format and are dependent on the selected ODBC database. You cannot change this column; you should use the Qnnn option in the field format for this.

7.4.5. C-byte

The C-byte is used for the internal program record and displays how the data is stored in a C-Structure. The information may be useful for developers.

No	Name	Format	Byte	CbyteCvar	Basicvar
1	Rectype	2,	0	0short	S1[1]
2	Debtor/creditor no	5SP	2	2char	S\$(1,5)
3	Name 1	30B8	8	8char	S\$(3,32)
4	Name 2	30	38	39char	S\$(33,62)
5	Street	30	68	70char	S\$(63,92)
6	Town	30	98	101char	S\$(93,122)
7	Country	30	128	132char	S\$(123,152)
8	Postal code	11	158	163char	S\$(153,163)
9	FC code, accounts	2,B170	170	175short	S1[2]
10	Gross/nett indicator	1,	172	177short	S1[3]

25. Byte, C-Byte, C-Variable and Basic variable

7.4.6. C-Variable

This option is useful for developers. It displays the C-variable type used internally for each field as char, short, long, float or double.

7.4.7. Basic-variable

When working with Basic programs it is useful to have the Basic variable available. For C programs such as TRIO this has no effect.

7.4.8. Record

If the table is accessible (able to be opened) it is possible to view the record contents together with the field definitions. Whenever you change a field definition this is reflected in this column giving a very fast check that the fields are defined correctly.

No	Name	Format	ByteRecord
1	CustID	5	0ALWAO
2	Company	40	6Always Open Quick Mart
3	Address	40	4777 Overpass Ave.
4	City	20	88Provo
5	Region	20	109UT
6	PostalCode	20	13084604
7	Country	20	151 USA
8	Phone	20	172(801) 555-7424

26. Record contents for table

You can toggle the display of the record contents on/off with this menu.

7.4.9. Next record

This function reads the next record from the table. If the table cannot be opened, e.g. illegal real name or wrong driver type, no values are displayed.

You also have this option available on the toolbar. Displaying next record automatically enables the display record option above.

The first time the function is selected it reads the first record in the table and then the next until end of the file.

7.4.10. Record overview

The function shows the first 100 records on screen, see 'Table information'.

7.4.11. Hexadecimal dump of a record

This function displays a raw dump of the record.

No	Name	Format	ByteHexadecimal
1	Article no	4B4K	30,31,30,31,00,
2	Description	20	43,48,4F,43,4F,4C,41,54,45,00,00,00,00,00,00,
3	Selling price	6,2	00,00,00,00,00,00,00,40,
4	Cost price	6,2	00,00,00,00,00,00,F8,3F,

27. Hexadecimal dump of a table

By use of this function a programmer may be able to crack problems in files using undocumented field packing or codetables.

7.4.12. Decimal dump of a record

If preferred the values may be displayed in decimal notation.

7.4.13. Ascii dump of a record

The record may also be shown as text limited by the display size.

7.4.14. Help overview

If any help exists for the fields this options displays the first line from the documentation giving a quick overview.

9	FC code, accounts	2,B170	FA134-137
10	Gross/nett indicator	1,	0 =gross
11	WORD indicator	2,	0 =no integration
12	Cash discount group	2,	0-40
13	Accounting date	2,	
14	Time allowed for payment	4,T1	MAX 7999

28. Help overview

7.4.15. Check overview

If check codes are defined for fields as many as possible are shown together with the field definition.

7.4.16. Special text overview

If special text is present for fields the first line of these is shown together with the field definition.

7.4.17. Reserved words

The function shows the SQL reserved words, see 'Table information'.

7.5. Amending DATAMASTER files using the FDF module

When you amend a DATAMASTER file using the FDF module the file itself may be kept unchanged while the definition is changed (careful!) as you are asked:



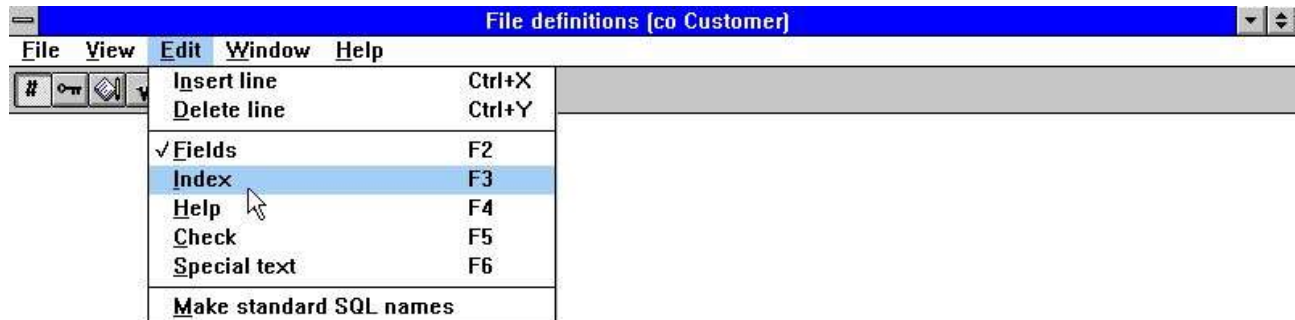
29. Confirmation of DATAMASTER file copy

7.5.1. DATAMASTER file conversion from BASIC to ODBC

When a BASIC file is copied to ODBC using DATAMASTER the informations of Packtype, Bytenumbers and Stopcharacter will now be removed as these would have bad influence on the resulting ODBC table.

8. Index, help and check definitions

From the EDIT menu you can reach the following areas:

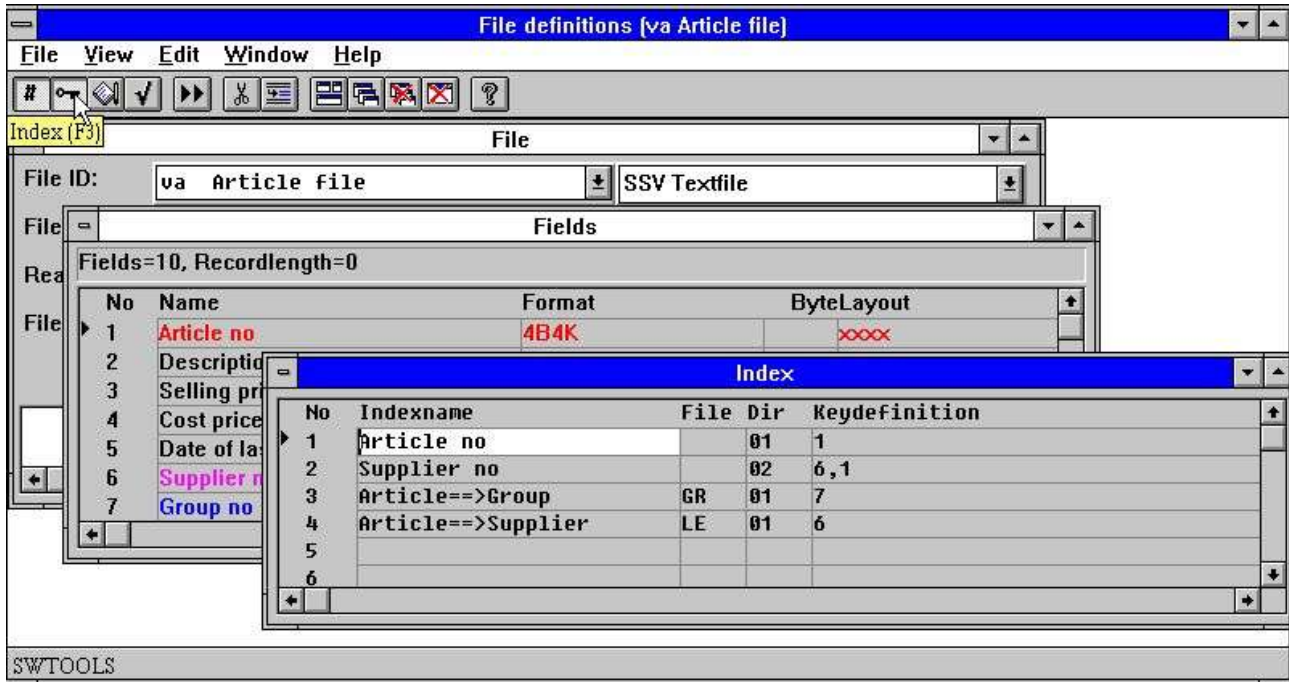


30. The Edit menu

8.1. Index definitions and table connections

Indexes define how the data is sorted in a table and how to access a specific record. Connections to other tables define how another table is accessed when a record in this table is read and is defined in the same way as an index.

To edit index/connections for a table select the following function:



31. Editing indexes

Indexes can be inserted and deleted using the following functions:



32. Inserting and deleting index lines

You may have noticed that some lines in the field window are marked in different colours. Red indicates that the field is used in one or more index definitions, Blue that it is used for a connection, Magenta (combination of Red and Blue) that it is used in both.

8.1.1. Indexname

The index name may contain letters, digits, blanks and special characters. It is normally used as description when SW-Tools TRIO displays a view of the table indexes, for example by starting a report printout.

8.1.2. File

The file ID is used only when defining relations from this table to others. When defining a relation this ID must refer to an ID of an existing table.

In the example above the article file has two relations. The first to the article group table in order to get the group name for each article. The second to the supplier table in order to get the supplier name.

8.1.3. Directory

The directory, also known as the index number, is a two-digit value.

For an index this is the index number starting with 01 also known as the primary key, 02 for the first secondary index etc.

For relations to other tables it defines which index to access when getting a record from the related table.

In the example above the supplier file LE is defined with a primary index, supplier number, when a record in the article file is known you can get the related supplier record by searching in the supplier table primary index with the value from the article file field 6, supplier number.

A special index is index number 00 which enables access to the file using record numbers. This can be used for file connections using relative record numbers when the database driver supports this, e.g. Basic systems.

8.1.4. Key definition

The key definition is the real definition of the index or connection. When defining an index this describes which fields are included and how they are stored.

A key definition may contain reference to one or more field numbers, part of fields and define field packing. An optional # may be given in front of the field numbers.

In the above example the first two lines define the primary and secondary index for the article file. The primary index refers to field 1, article number, defining that the table is sorted by article number in ascending order. The secondary index is made up of two fields, field 6 (supplier) and field 1 (article number).

This permits access to the table using supplier number and enables printing of lists sorted by supplier number and article number and allows more articles to have the same supplier as the combination of supplier number and article number will always be unique.

Lines 3 and 4 define relations to other tables. The key definition refers to the values needed to access the index of the related table. The first relations use the value of field 7, article group number, to access the article group table index 1. The second uses the field 6, supplier number, to access the supplier table index 1.

8.2. Index considerations

Above was shown simple key definitions just involving plain fields and these in combination. Each field of an element in an index definition is known as a key part.

How many indexes you can define for one table, how many parts an index can consist of, how large the limit of the total keylength is, and which parts and packing may be used are strictly driver dependent and you should refer to the appropriate database system manuals for exact information on this.

Basic allows a maximum of 15 indexes of a maximum of 30 bytes each, as the key is always treated as a character string, there is no limit of the number of key parts that can form the key as well as part of fields can be included in the key. All keys must be unique or made unique. A special feature is that the key is not necessarily stored in the record itself.

Isam specifications vary, but may typically be 128 indexes with a maximum length of 128 bytes and up to 8 key parts for each. Each key part must be a complete field and must be present in the record. Duplicate keys can be defined and index compression may be specified.

ODBC SQL database systems use index keys to optimise performance but a new key must always be built using an ORDER BY which may slow down the retrieval of records.

In order to support the different database systems a number of options may be given together with the key definition. All options may be used with all database drivers but some are meaningless and may cause error messages if the driver does not support the functions.

8.2.1. Basic COMET AF-05000 example

Lets first have a look at the COMET AF-05000 sales order lines in Basic:

The screenshot shows the SWTOOLS File definitions window for the file 'au Sales order 20/ Item reco'. The window is divided into several panes:

- File:** File ID: au Sales order 20/ Item reco, XBasic; Filename: AF-05000abc; Standard ID: (empty).
- Fields:** Fields=34, Recordlength=160. A list of fields is shown:

No	Name	Format	ByteLayout
1	Order no	6P	0>xxxxxx
2	Rectype		
3	Item no		
4	Continuation no		
5	Status		
6	Article no		
- Index:** A table showing index definitions:

No	Indexname	File	Dir	Keydefinition
1	Order no+Item no		01	1,"20",3,4,P
2	Planned+Order		02	16,1,"20",3,4,P
3	Recno Arti+Planned		03	"20",33,16,1,3,P
4	-->Auftragskopf	AS	01	1,"00000000",P
5	-->Artikel	D4	00	#33,N
6	-->Lagerort	D7	00	#34,N
7				

33. AF-05000/020 Sales order lines index and connections

8.2.2. Constants and selection

1,"20",3,4,P

A constant may be given as "20" as in the definition of the first index which consist of field 1 followed by the constant (record type) "20" and then field 3 and 4.

Definition of a constant in an index triggers a selection when the file is read. There may be keys with different values in the constant key positions but only record which matches will be returned.

In COMET such constants are often used for a record type where completely different records are stored in the same file. Each record type should be defined as a separate LOGICAL file with connections to the other record types.

8.2.2.1. Multiple constants in index definitions

Files defined with fixed constant recordstypes in index as "00",#1,P may now be extended to "00,20-29,40",#1,P

The selection will then retrieve all records with one of these constants.

8.2.3. Packing

1,"20",3,4,P

All key parts in front of the ,P here will be packed. The P option could also be used for single fields as 1,"20",3P,4 where only field 3 is packed.

As mentioned the packing is specific for each database system, for Basic CALL 60 is used.

8.2.4. Record numbers

#38,N

The record number for the current file can be referenced by R, the relative record number by N. These may be used just like fields to form keys and connections to other files.

8.2.4.1. Index 0

#38,N

As mentioned index 0 means reading the file using the record number not through a index. When you use index 0 the key definition is treated as a calculation which means a # must be inserted in front of any fields and a syntax like #38+1+N is allowed.

8.2.5. Duplicates allowed

#17,NP

An index can normally hold one occurrence of a value only, i.e. is the key unique. For some indexes it must be possible to have several records with the same key value.

A common way in Basic is to add the relative record number packed (,NP) after the field which make the total key unique.

#17,M

For ISAM systems ,M may be stated for Multiple values as index may be stored directly in the database system with duplicates allowed.

8.2.6. Zero suppress

#17Z

Basic numeric fields are converted into a text string to form the key, the standard conversion will place leading zeroes in the fields. By use of the Z option leading blanks are inserted instead.

8.2.7. Space fill

#17,S

S causes the text string corresponding to the key to be padded with blanks. This has only effect if the file is built with a key longer than the number of characters in field 17.

8.2.8. Part of fields

#17(2,4)

You may take position 2, through only 4 of field 17 in this way.

8.2.9. Key not in record (KNIR)

In Basic the key does not have to be included in the record. Such keys are defined in exactly the same way as all other keys but the fields referenced should be marked with W in the format.

Using this the index values are placed in the fields during a read of the file and the fields may be used just as normal fields for printing or for connecting to other files.

However note that KNIR W fields defined on index 1 is filled only when index 1 is used as access key for the file. If index 2 is used only the key from index 2 is known and only fields attached to this index can be filled.

8.2.10. Index key definitions, LOGICAL index

The meaning of index / directory numbers has been worked over.

Whenever you refer to an index for example by selecting this by start of a report of by reading a file using READ(xx.NN) you refer to the LINE NUMBER in the index definitions.

The LINE NUMBER in the definition now becomes a LOGICAL INDEX NUMBER which may or may not reflect the physical directory stored in the files.

As the index lines previously was sorted according to the directory number of the file all existing definitions have LINE=DIRECTORY so no backwards compatibility problems will occur.

This sorting has been removed for all other than DATAMASTER files enabling you to define the indexes in which order you want. The index order is used for example when searching in IQ.

The DIRECTORY number stated on a key definition line is for use by the database driver only for the physical search in the directory. It has nothing to do with the INDEX number NN stated in READ(xx.NN)

One DIRECTORY may occur multiple times as LOGICAL index lines defining different ways of building a searchkey. This is sometimes used in BASIC files as an example you could state different "XX" constants as logical index without having to duplicate the complete file definition.

8.2.11. Directory 00 recordnumber reading

Also you may now define an index connecting to directory 00 also, which means record number reading. This index may be selected by start of a report just like others and you may even define a KNIR field (Key Not In Record) becoming the recordnumber when this index is used.

8.2.12. ACCESS using SWODBC on non-indexed files

Microsoft ACCESS do have problems accessing any file without index defined. Defining index 00 as above overcomes this problem.

8.2.13. Suppressing IQ search on specific index (*xx)

One or more index may be omitted from the standard IQ search by stating the directory number as *XX. The index is skipped unless you specifically select this to be searched in the IQ program.

8.2.14. Suppressing index update for a specific index (-xx)

Stating -XX as directory number courses BASIC files not to update the file according to this index definition. May be used when defining multiple index for a file.

8.2.15. Creating a pseudoindex for a file (+xx)

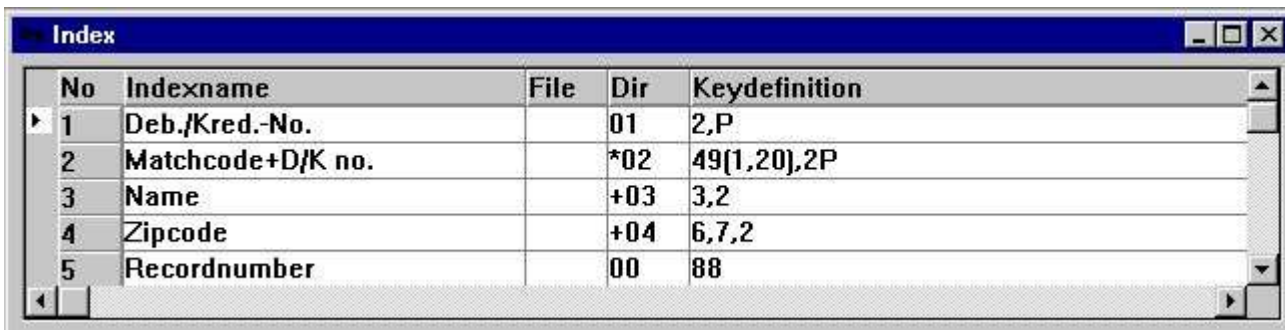
- THIS FUNCTION IS FOR THE TIME BEING NOT FINALLY RELEASED -

Stating +XX as directory number, where XX is a non-present physical directory in the file, causes the extended SSV interface (see this) to construct the index as needed. Always set XX=Line number so the physical constructed index becomes equal to the logical index used.

The index may be used just as it was physically present. In case of use of a server the index is constructed here, not on the local PC.

Due to the complexibility of this function it will not be finally released in this version of TRIO as very extensive tests are needed. However you may find this testrelease useful.

8.2.16. Index example



No	Indexname	File	Dir	Keydefinition
1	Deb./Kred.-No.		01	2,P
2	Matchcode+D/K no.		*02	49(1,20),2P
3	Name		+03	3,2
4	Zipcode		+04	6,7,2
5	Recordnumber		00	88

34. Example of pseudoindex and recordnumber definition

8.2.17. Key options, Descending index

A couple of keyfield options has been added:

I = Invert all bits in the keyfield (Descending index)

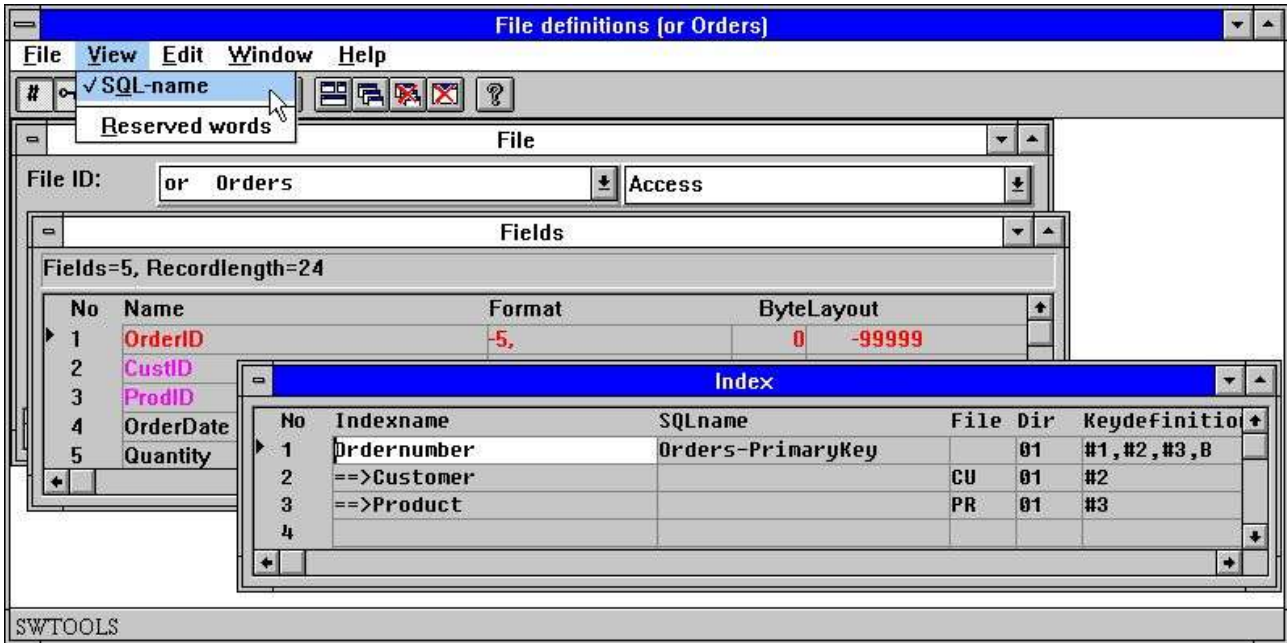
C = Global files only, Compress DMAM index

S = Global files only, Do not invert the sign bit of numeric field

An index definition as: 5I,2 will invert all bits in field 5 thus defining a descending index.

8.2.18. ODBC Access orders example

The Access orders table definition could when imported look like:



35. Access orders table with SQL names

8.2.19. Index SQL-name

In an SQL Database each index has a name which must be a valid for the driver. This name is used only when creating the table or redefining it, for normal reads it does not have to be stated.

From the VIEW menu during index definition you can add the column with index SQL-names and enter the names yourself. By generation of SQL-Names a standard name is assigned to all indexes.

8.2.20. Born indexes and ORDER BY

#1,#2,#3,B

this defines an index consisting of fields 1,2 and 3. The B tells the index is Born on the table and no ORDER BY is necessary. If the B was omitted, the SELECT when reading the table would be added an ORDER BY with the stated fields.

8.2.21. Descending indexes

#3,D

By using D a descending index instead of the default ascending index can be defined.

8.2.22. Special code table

#2U2

U2 causes code table 2 to be used before submitting the field to the index.

8.3. Table connection considerations

Relations between tables are defined like indexes.

Please note that the application program will use BOTH the relation definition and the index definition for the file that is to be read.

The relation will be made by filling the field in the index definition with the fields stated in the relation definition. This means that in case of problems in getting a specific relation to work both definitions should be checked.

8.3.1. Variable length keys

File connections dependent on the key length #3(1,-2) has been implemented, #3(1,-2) meaning field 3 from character 1 to the keylength-2 characters.

8.3.2. X* Fixed parameter file connections

A parameter file which should read a fixed record always may now define a connection which are used by all READ(pa) when no other connection is given. Definition may be done as a connection from PA to file X* index 0 recordnumber 5

8.3.3. File connection definitions

As any READ(xx.NN) now lets NN reference the LOGICAL INDEX LINE number this is the case also when defining file connections inside the FDF.

For a connection in the column DIRECTORY you state the LOGICAL INDEX LINE which should be used to read a file.

8.3.4. Multiple connections to the same file using upper/lower case

READ(xx) searches for a connection to file xx stated in lowercase first, if not present the first connection found is used regardless of case. XX, Xx, xX may be used to state different connections.

The sorting of file connections in the FDF has been changed so all connections to a file is shown together regardless of case.

8.3.5. Multiple connections to the same file using logical index

READ(xx.02) will use the connection given to LOGICAL index 2 of file XX. This may also be used to state more connections with different fields.

8.3.6. Forcing connections by READ

READ(aa.bb) will read the file bb using a connection from aa->bb.

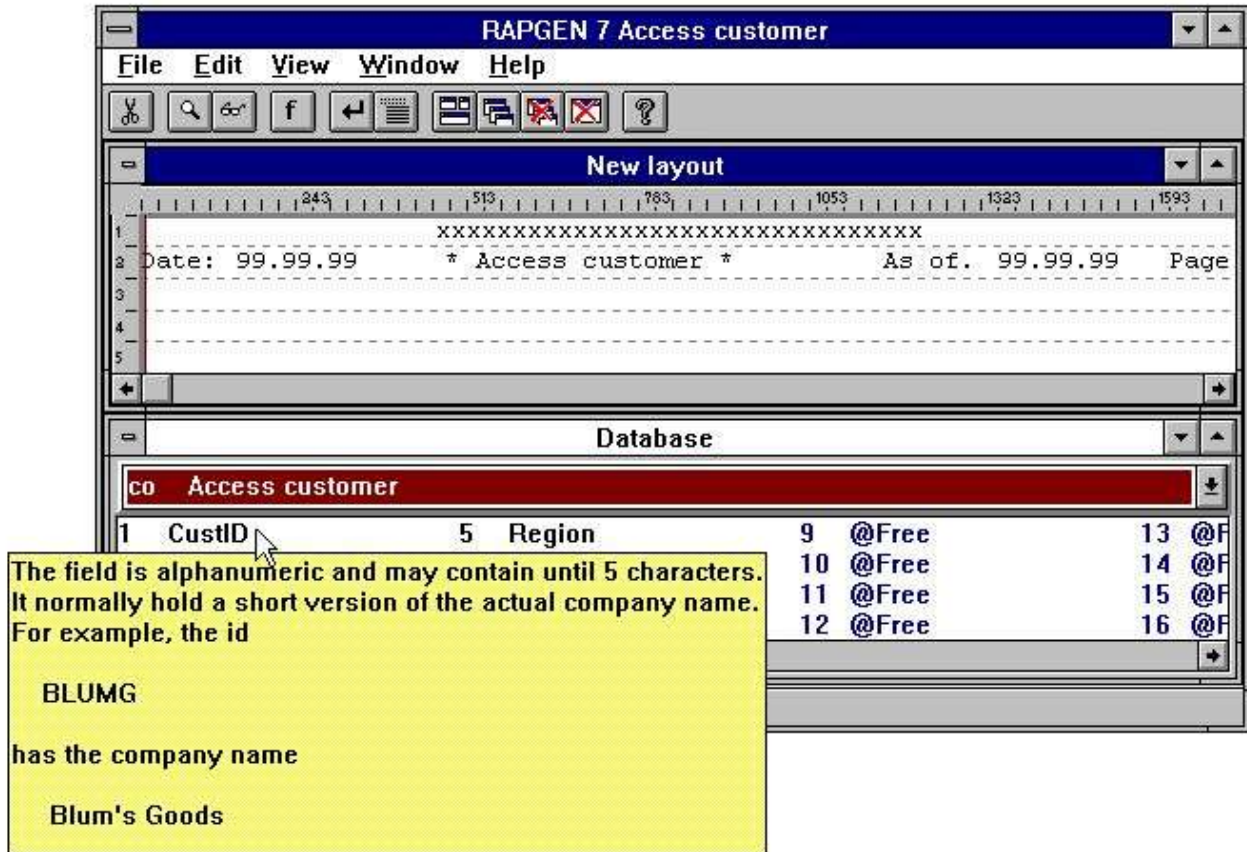
READ(aa.bb.02) will read the file bb using a connection from aa->(bb logical index 2)

8.4. Field help descriptions

No Data-Dictionary is complete without documentation. It is possible to write documentation for the table and for each field. When documentation is written for a field it provides a feature in the TRIO package.

Floating on-line help in field overview

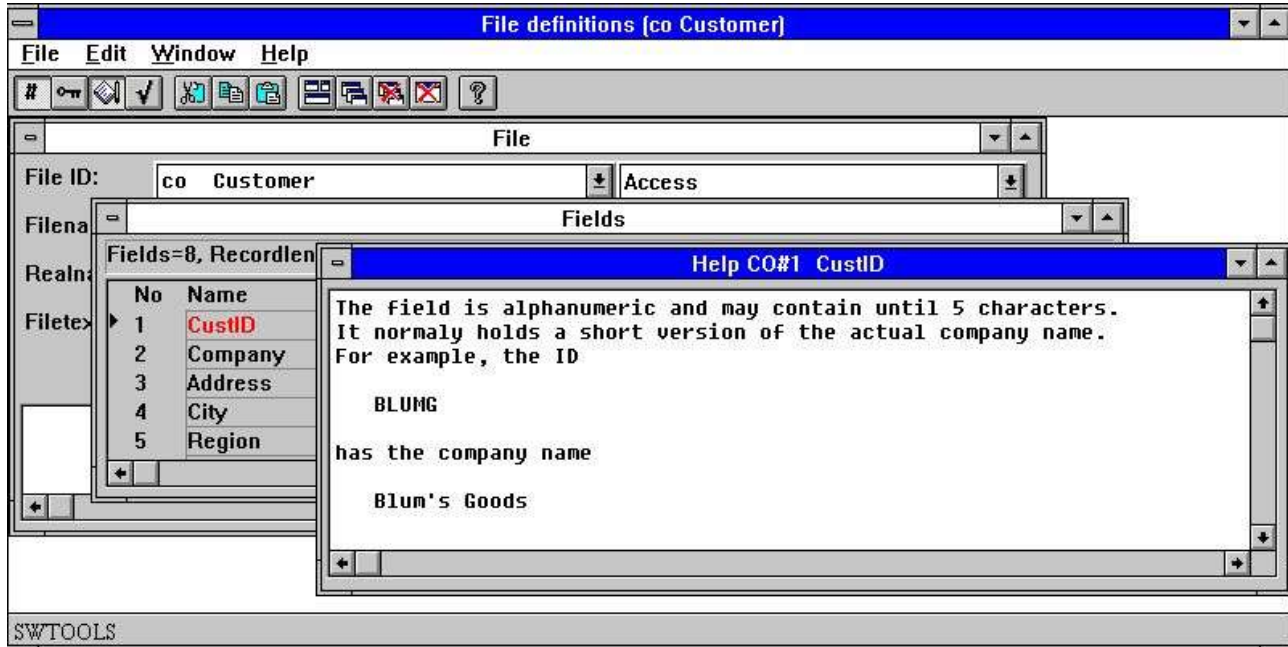
When using the SW-Tools TRIO Report generator, IQ enquiry application and the DATAMASTER the database window will display the floating help when the cursor is over the field. For example in RAPGEN:



36. Floating on-line field help

8.4.1. Editing field help description

When editing fields the function has a marker for the current field. To edit the documentation for a field, first make sure the marker points to the required one, then select HELP from the EDIT menu:

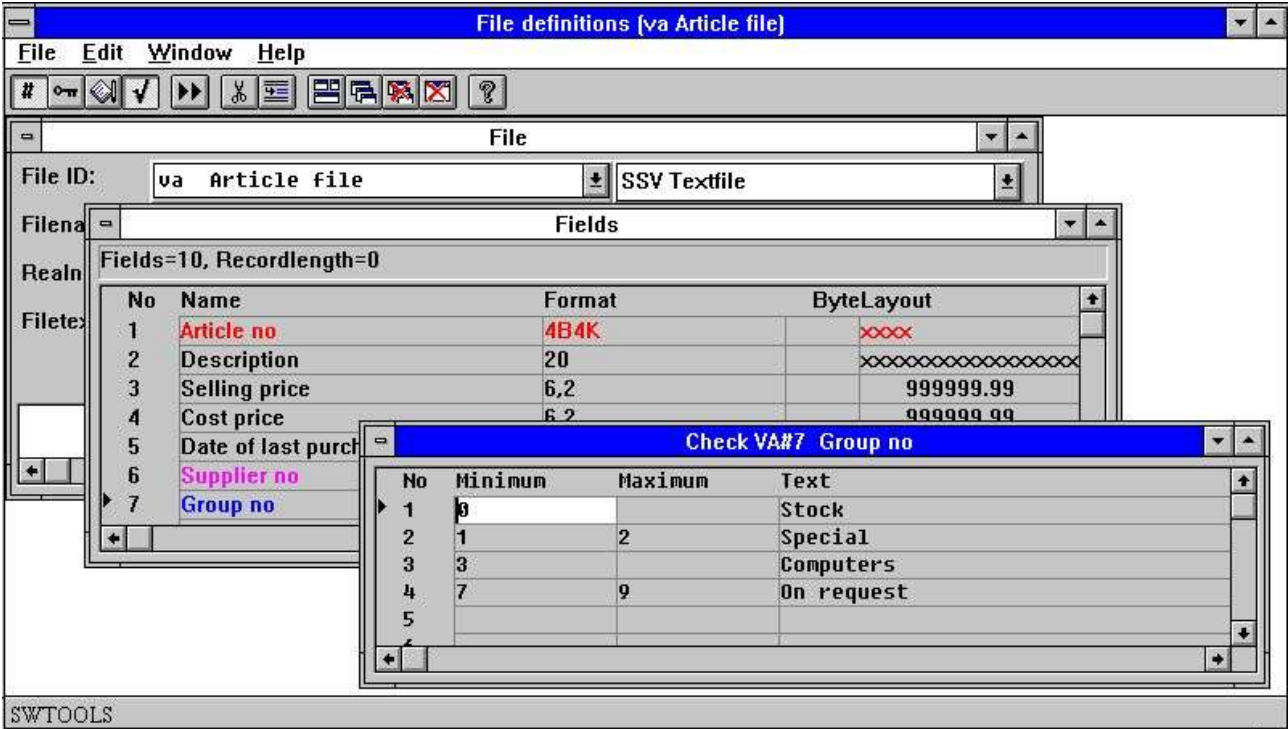


37. Editing field help description

8.5. Field check codes

Field check codes are used to define allowed values for fields. This is used extensively in SW-Tools DATAMASTER for input validation of fields, and may also be used in TRIO Report generator/IQ to associate a text with a code field.

Field checks are defined as a number of ranges from-to value and an associated textline.



38. Editing field check

8.6. Field special text

The special text associated with a field is just a second help page, the input is done exactly as described for the normal help.

These texts are reserved for additional field parameters and calculation but not activated.

However if you import definitions from our COMET file definitions you may get special text with some READ calculations and definitions of how the parameter files should be read by installation.

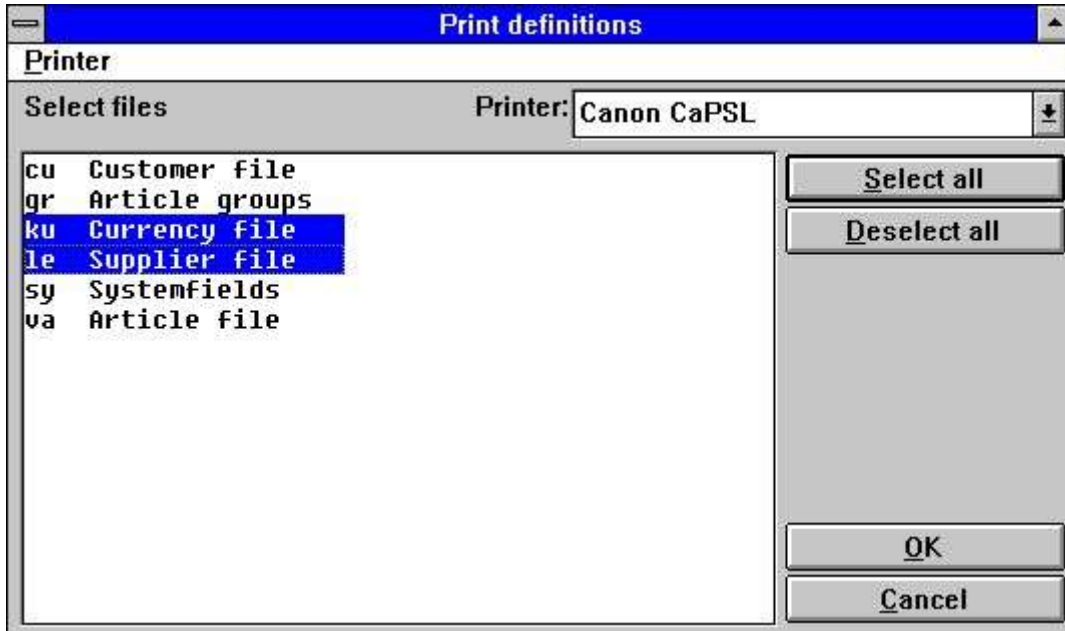
8.7. Make standard SQL-names

This function may be used if you create a new file definition for an SQL database using this module.

If you use the TRIO DATAMASTER SQL, the names are all automatic and when importing table definitions from ODBC databases the SQL-names are taken from the drivers table definitions.

9. Printing documentation

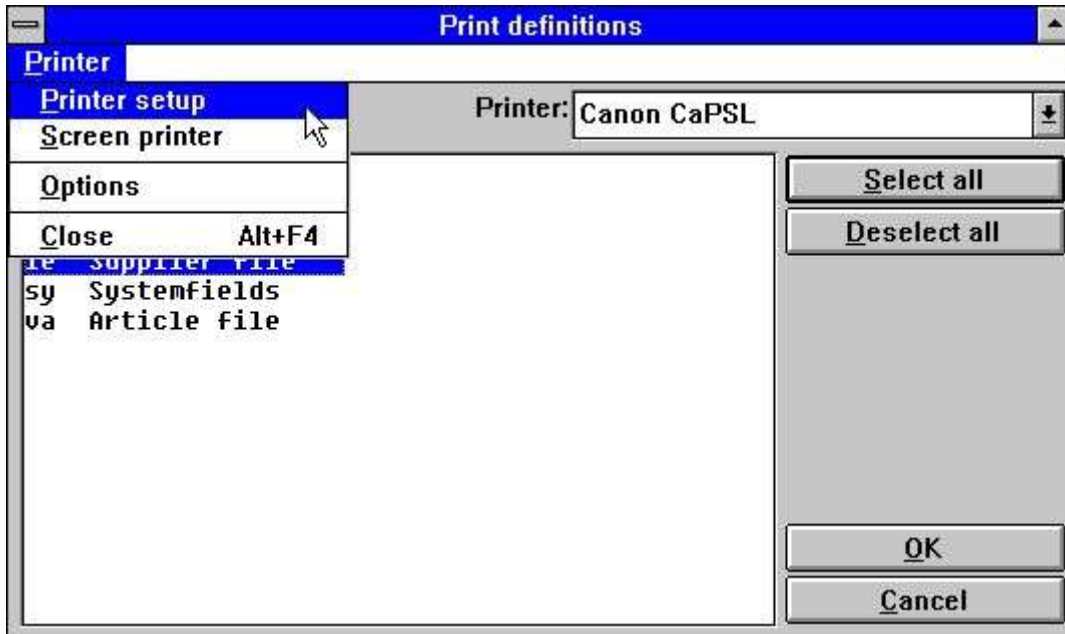
To print documentation on table definitions use the following function and select one or more tables:



39. Printing documentation

9.1. Printer

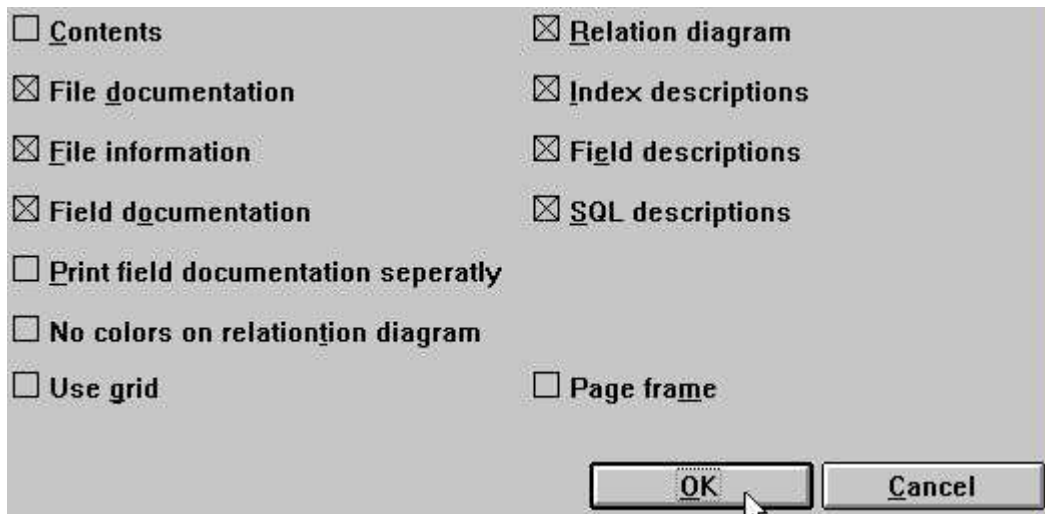
The documentation can be printed on any Windows installed printer. It can also be printed on a screen printer.



40. Selecting printer

9.2. Options

When printing the documentation it is possible to control the detail level of this:



The image shows a dialog box with a light gray background and a blue border. It contains two columns of checkboxes. The first column has: Contents, File documentation, File information, Field documentation, Print field documentation separately, No colors on relation diagram, and Use grid. The second column has: Relation diagram, Index descriptions, Field descriptions, SQL descriptions, and Page frame. At the bottom right, there are two buttons: 'OK' and 'Cancel'. A mouse cursor is pointing at the 'OK' button.

41. Documentation options

Data Dictionary (006.006) SW-Tools Page 1

va - Article file

This is the standard demo article file.

The file is present as SSV file and in BASIC systems.

Name:
Driver: SSV Textfile
Fields: 10
Index: 2
Relations: 2
Relation diagram:

```

graph LR
    VA[VA Article file  
Article no  
Supplier no  
Group no] ---|1: Group no| GR[GR Article groups  
Group no]
    VA ---|1: Supplier no| LE[LE Supplier file  
Supplier no  
Currency code]
    
```

Indexname	File	Dir	Keyfields
Article no		01	1
Supplier no		02	6, 1
Article==>Group	GR	01	7
Article==>Supplier	LE	01	6

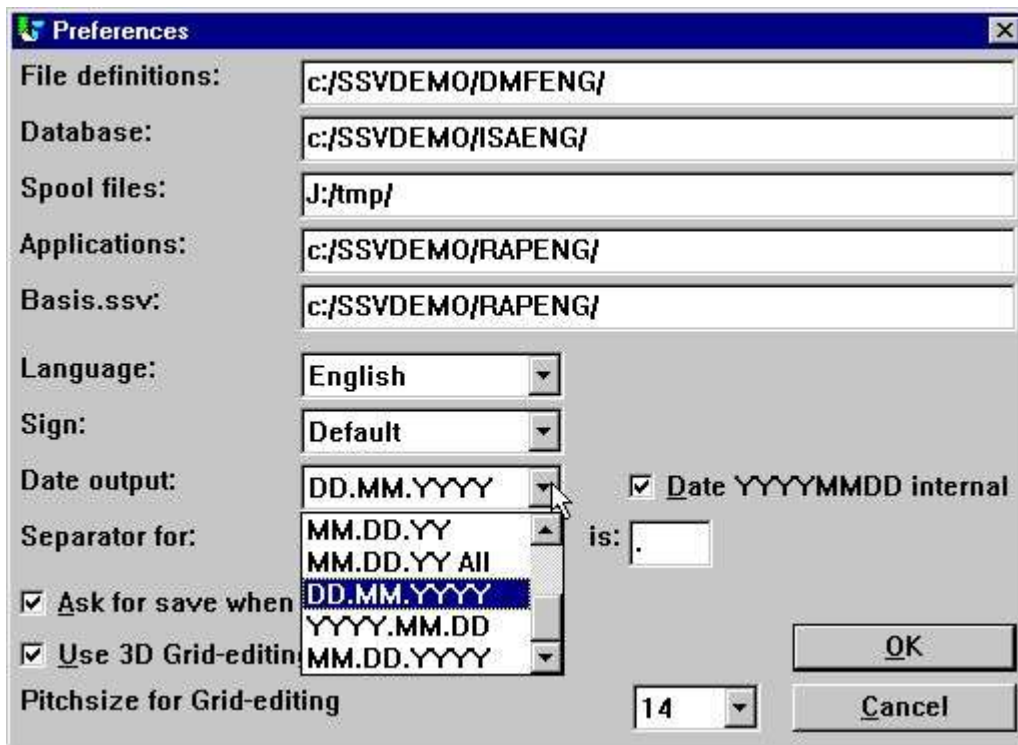
Field name	Format	Layout
1 Article no	4B4K	xxxx
<i>The article number is the main key to the file.</i>		
<i>A maximum of 4 digits can be entered.</i>		
2 Description	20	xxxxxxxxxxxxxxxxxxxxxx
3 Selling price	6,2	999999.99
4 Cost price	6,2	999999.99
5 Date of last purchase	, 6,	99.99.99
6 Supplier no	3	xxx
7 Group no	2,	99
0 = Stock		
1 - 2 = Special		
3 = EDP		
7 - 9 = Not on stock		
8 Stock balance	-6,	-999999
9 Alternative supplier	3,	999
10 Free	-6,2	-999999.99

42. Example of table Documentation

ID	Description	Name	Standard ID	Interface
d4	Artikel-Stammdatei 1	LF-06000abc	LF-06000 4 D4	XBasic
d7	Lagerort Datei	LF-06001abc	LF-060010204 D7	XBasic
db	OPEN ITEMS FILE	03/GF-03100abc	GF-03100abc	xnet
dk	DEB/CRED MASTER FILE	03/GF-TEST	GF-TEST	xnet
dx	CREDITOR MASTER FILE	03/GF-03000abc	GF-03000abc	xnet
ex	MAPPE1	MAPPE1		Excel
gr	DEMO-ARTICLE GROUPS	03/DMS.F..GRENG	DMS.F..GRENG	xnet
jh	Debtor/creditor master 1	3/gf-test	GF-03000 2 GM	xnet

43. Example of table of contents

10. Preferences



44. Preferences

10.1. Directories

All SW-Tools applications uses the initialisation file RAP.INI located in the startup directory to locate all other application files.

The startup directory is given in the Windows icon. Different icons with different startup directories may be created enabling for example several different Data-Dictionaries on the same PC.

This file contains directory paths for the following:

10.1.1. File definitions

Define the directory path for the Data-Dictionary itself. When defining a table in the Data-Dictionary the system generates SSV-files as

<table id>.SSD

and updates an overview file

FILES.SSV

The first file contains all information of the single table defined. The second is used only for displaying file overviews.

10.1.2. Database

This path is the standard database file location for all drivers. If a driver needs to locate its data files through a path and no path is defined for the driver the path given here is used. It is the path of the physical location of the tables in the database.

10.1.3. Spool directories

When anything is printed or generated the system requires space for temporary and spool files. This path should be set to a directory where enough space is available for print files. If left blank the environment variable TMP or TEMP is used.

10.2. Language

English, German and Danish are supported at present. However your master installation disk usually comes with one language only. If more are needed please contact SW-Tools.

10.3. Sign

The sign option is used when numeric field values are printed or displayed. It has three options:

- **Default**
- **Leading**
- **Trailing**

The default options allows definition of a format as

-9,2

and hereby the sign is leading. But if defined as

9,2-

the sign is trailing.

The other two options do not look at how the format was defined. It simply sets the sign to leading or trailing.

10.4. Date output format

Date fields can be formatted as Day/Month/Year or Year/Month/Day. The default option is in the format Day/Month/Year.

Possibility to output dates as DD.MM.YYYY and the American way round MM.DD.YYYY has been added:

```

0 = No conversion
1 = ,6,          -> DD.MM.YY
2 = ,6,          -> YY.MM.DD
3 = ,6, and ,8, -> DD.MM.YY, ,8,& -> DD.MM.YYYY
4 = ,6, and ,8, -> YY.MM.DD ,8,& -> YYYY.MM.DD
5 = ,6,          -> MM.DD.YY
6 = ,6, and ,8, -> MM.DD.YY ,8,& -> MM.DD.YYYY
7 = ,6,          -> DD.MM.YY, ,8, -> DD.MM.YYYY
8 = ,6,          -> YY.MM.DD ,8, -> YYYY.MM.DD
9 = ,6,          -> MM.DD.YY ,8, -> MM.DD.YYYY

```

Note the new options as DD.MM.YY-All with the possibility to override the ,8, YYYYMMDD output also.

A ,6, field will for nice output editing be threatred as ,6,&' which means leading zeroes and zero suppression.

10.5. Separators

When numeric fields are edited the system needs three separators:

- **Decimal point**
- **Thousand edit**
- **Date edit**

By changing the separators it is possible to change the output of a numeric field from being

123,456,789.12

to

123.456.789,12

10.6. Ask for save when changes in report

When this box is checked you must confirm any changes made to file definitions, reports or IQ program.

If not checked the amendments are written without further notice.

10.7. Internal date calculation format

- BE CAREFULL BEFORE YOU ACTIVATE THIS OPTION -

If your system stores dates in the short format YYMMDD you have had to take care of this if you calculates on date fields in all reports and IQ programs.

This may have been done with calculations on a ,6, field #47 like:

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

If you activate the YYYYMMDD calculation option all shortdates will be converted by reading(writing) the files to long format YYYYMMDD year 19xx/20xx.

You will have to change your calculations after activating this:

IF #47>#DD LET #99=1

It is of course preferable that the reading does the date conversions itself but great care should be used when activating on live systems.

10.7.1. Year 2000

As TRIO is integrating to many different file systems all sort of year 2000 patching and tricking are seen.

Among others support for X-Basic COMET dates as alpha plain or packed in index keys has been added as described under packtypes.

10.7.1.1. Date sorting

If you sort on a shortdate ,6, field the sortfile will be builded with ,8, dates. The same rule is applied on grouptotals in the report generator.

Year 2000 is also sorted correctly by use of ORDER BY on any ,6, shortdate field.

10.7.1.2. DOS dates

DOS and equivalent systems will return 100 for year 2000.

A date field of ,6, or ,8, with value between 1.000.000 and 1.999.999 will be assumed to be in year 2000.

10.7.1.3. Year 2001 problem

Note that the FNY function will have a problem if you passes DDMMYY dates:

FNY(31.03.98) = 1998.03.31 ok
FNY(31.03.00) = 2000.03.31 ok
FNY(31.03.01) = 2031.03.01 YYMMDD is preferred instead of DDMMYY

10.7.1.4. Year 2005 problem

In BASIC packed keys the first packed character is handled as year 2000 when not a digit. 5 such characters are available: space,plus,comma,minus,point reaching from year 2000-2004 only.

10.7.1.5. Year 2010 problem

In BASIC unpacked keys the first character is handled as year 2000 when this is a question mark (?) enabling year 2000-2009.

TRIO will handle ?@ABCDEF... as year 200x,201x,202x,... also.

10.7.1.6. Year 2100 problem

When the year is added to a shortdate YYMMDD year 2000 is assumed when YY is less than 50.

When the system date passes year 2050 the same rules will be applied just for next century.

10.7.1.7. Date input validation

By input of date fields the following rules now applies in the stated order:

0. Input 0 -> 0
 1. Input DD -> DD+MMYYYY from system date
 2. Input DDMM -> DDMM+YYYY from system date
 3. Input DDMMYY -> DDMMYY+TT from system date as above
 4. Input DDMMYYYY -> DDMMYYYY, is accepted in ,6, fields also
 5. Date validation according to preferences
 6. If fails, Date validation as: DDMMYYYY, YYYYMMDD and MMDDYYYY
- .

10.7.1.7.1. IQ selection date input

When you use the transaction selection input in IQ for date fields, your input and the date field will be passed the FNY function always to ensure the year is set.

10.7.1.7.2. CHAIN function masked date input

When you use CHAIN to start a report the dates you may pass by the function call will be threated as input just as by manual start of the report.

10.8. FDF Start parameters

The FDF module may be selected from WINDOWS using the following parameters:

```
RAPFDF -d      Files will be defined as DATAMASTER files  
RAPFDF -d7    Datamaster default filetype will be no.7
```


11. Subsystems

Using the Preferences menu you define a SYSTEM by setting parameters in the RAP.INI file on the current working directory. This file is read by program start.

A SUBSYSTEM can be selected during the program run. The subsystem setting then replaces or modifies some of the system parameters.

A typical use could be grouping of report or IQ programs. You can do this by duplicating the report generator icon in Windows with another working directory (it is a new RAP.INI) or just by creating a subsystem pointing to another directory for the reports.

11.1. The subsystem menu

Subsystems are selected and created/amended and deleted from the subsystem menu:

<u>P</u> references...			
<u>S</u> ubsystem		<u>O</u> pen...	
<u>C</u> lose	Ctrl+F4	<u>N</u> ew...	
		<u>C</u> hange...	
		<u>D</u> elete	
		<u>S</u> ystem files	
		<u>C</u> ompany file...	
		<u>C</u> odetables...	

45. The subsystem menu

11.2. Opening of a subsystem

This function actually selects a subsystem for use just as the program selector in Rapgen or IQ.

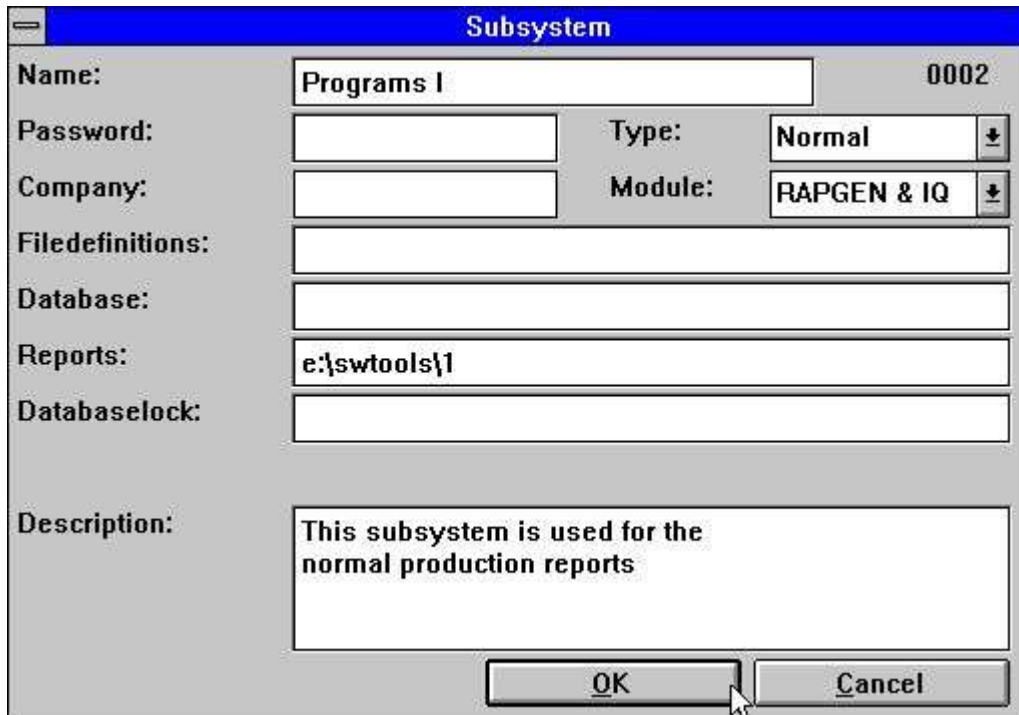


46. Opening subsystems

Subsystems work as folders, you open/close these by clicking on an icon or name. Each subsystem can contain one or more entries. An entry can be a program or an item fixing some information for the next program as company number.

11.3. Creating a subsystem

The functions New, Change and Delete subsystem work with either subsystems or single items within a subsystem dependent on the active selection when you invoke the function. A single program in the selection may also be modified.



Name:	Programs I	0002
Password:		Type: Normal ↓
Company:		Module: RAPGEN & IQ ↓
Filedefinitions:		
Database:		
Reports:	e:\swtools\1	
Databaselock:		
Description:	This subsystem is used for the normal production reports	

47. Defining a subsystem

Normally only a few of the fields are filled for one subsystem. Fields left blank will be taken from the standard preferences or may be set by another subsystem.

11.3.1. Name

You may enter free text here which is displayed together with the subsystem icon.

11.3.2. Password

If a password is stated this must be entered when you select the subsystem.

11.3.3. Company

A company number can be entered as 001 or 444. This controls which files are opened, see later.

11.3.4. File definition path

Each subsystem may have its own file definitions in which case the path should be stated here. If left blank the standard file definitions (set by Preferences) are used.

11.3.5. Database standard path

This field overwrites the Database path set by preferences. This path is normally used only if the database resides on a PC.

11.3.6. Reports directory

The subsystems may be used to split several reports into logical parts in which case each must have its own directory.

11.3.7. Databaselock

This field forces all files to be a given database type and should normally be left blank.

Your files may be defined as X-Basic files, X-Basic being file type 2 (second line) in the BASIS.SSV database interface file and connected to server 200.0.0.1. If your Database lock states:

2=3

file type 3 will be used instead of file type 2 for all files. This could then use another server 200.0.0.2 or even another file system such as XNet.

More assignments can be included:

2,7-8=3,9=4

or all types can be forced into one type

***=3**

11.3.8. Description

Free text may be entered here. This will be displayed when moving the mouse over the subsystem name in the selector.

11.3.9. Module

You can select whether this subsystem should be available for Rapgen only, IQ only, or both.

11.3.10. Type

Subsystems are divided into two different types, Normal and 'Sticky'. When you leave a normal subsystem all the information from it is deleted, as opposed to a 'sticky' subsystem where some information is retained.

The normal subsystem operates like program folders. When you select another folder you get a new set of programs in a new directory.

The 'sticky' subsystem may be seen as parameter selections for other programs. If you are working with different companies these may be created as 'sticky' subsystems. When you then select a company this will be used for all programs until you select another company.

Again the sticky subsystems may be divided into different categories dependent how you want to use these:

No	Name	Usage	File
0	Normal	Not sticky	-
1	Company	Modifies the company number	COMPANY.SSV
2	Datadict	Selects another data dictionary (file definitions)	DATADICT.SSV
3	Reports	Selects another report directory	REPORTS.SSV
4	Datasyst	Selects another file type for files in this data dictionary	DATASYST.SSV
5	Basis	Selects a file type from BASIS.SSV and fixes this for all files	BASIS.SSV
6	Dmsystem	Selects a subsystem from the subsystem file	DMSYSTEM.SSV
7	Standard	Enables modifications in the standard New report and New letter	-

11.3.10.1. Company subsystem type

When you select an item in a 'sticky' subsystem no program is started but instead a message telling this is activated is shown:



48. A company selector

The parameters set for the subsystem will now be active when you select reports/programs or other subsystems unless such a selection redefines these. In this case company 004 is activated and the company number in all other subsystems should be left blank so as not to overwrite this.

11.3.10.2. Datadict, Reports and Datasyst subsystem types

You can make each subsystem point to its own data dictionary, reports or database interface or you may define a 'sticky' subsystem as shown above for companies. The only difference is which file holds the information.

11.3.10.3. Basis subsystem type

This is a 'sticky' selection of one fixed database interface from the BASIS file. The information in here should be modified from the 'Database interface' menu not from the 'Subsystem-Change'.

11.3.10.4. Dmsystem subsystem type

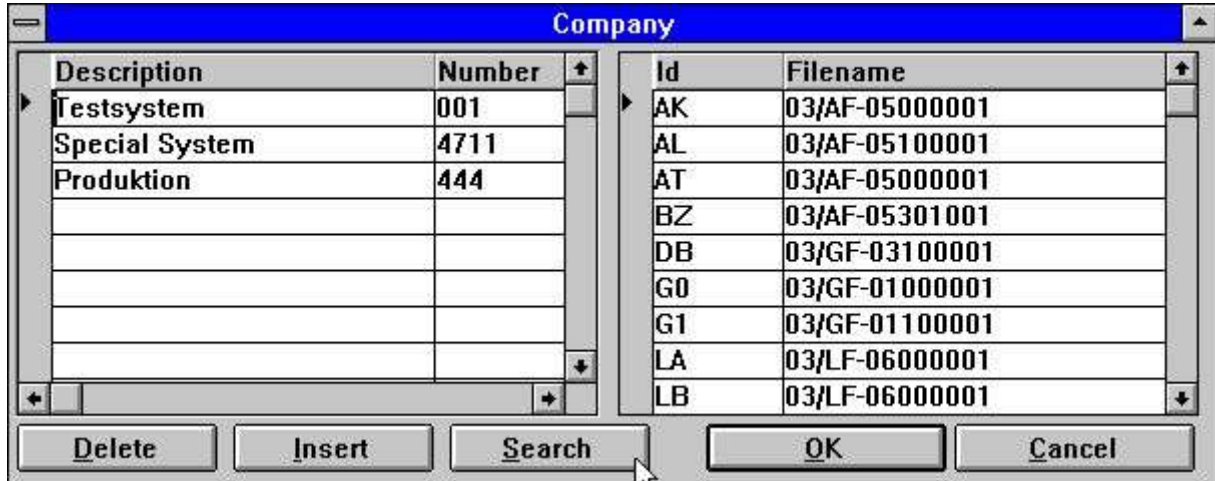
A normal subsystem is not sticky - the parameters are forgotten when you make another selection. From here you may select a normal subsystem but the parameters will become sticky.

11.3.10.5. Standard subsystem type

In here you will find the standard headings used with a new report or a new letter. These may be tailored with your company name or other information.

11.4. Company filenames

In here on the left side you may define name/number for the companies. This can also be done from the subsystem menu and goes into the same file (COMPANY.SSV).



49. Company filenames

When you click on a subsystem the list on the right side opens enabling input of a filename specific to each file in this company. These filenames are stored in <company>.KOM, e.g. 001.KOM, enabling use of different files for each company.

11.4.1. Company filename search

When installing different company on BASIC systems you have the possibility to search all LU's for the files by pressing the SEARCH button.

All BASIC files entered as 2-digit lu + filename, e.g. 90/GF-03000abc, will be checked if present on lu 00-99 in the selected company. When found and if the lu differs from the stated lu the file is inserted in the company filename table.

NOTE: Server version (006.003) minimum is required for this function.

11.4.2. DOS filenames for BASIC files

Working offline the server with X-Basic systems causes problems with the length of the filenames. It is now defined that if a basic file is not found on a DOS system a filename will be made as

- 1. All points are removed**
- 2. If more than 8 characters a point is placed after position 8**
- 3. The name is cut down to 8.3 characters**

This enables you to have a server file GF-03000001 local as GF-03000.001

11.5. Codetables

A codetable is stored in the file CODnnn.LLL, e.g. COD013.ENG, nnn being the codetable number and LLL the language.

Codetable 000-009 are reserved for internal hardcoded tables.

Codetables may be stored in binary or in a plain ascii textfile in which case these must be marked with the text SW-Tools first, on the first line and may be edited with this function. If a character should not be converted it may be left blank in the codetable which gives a much better overview:

```

C:\RAPMDI\COD049.ENG
File Edit
SW-Tools* 049 German BASIC parity 24/05-9
* Indicates default blank codes reverses parity

Addr. 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDE
000000: .ü..ä.ä.....Ä
000010: .ææ.ö....öüø.ø.
000020: !"#$%&'()*+,-.
000030: 0123456789:;<=>
000040: @ABCDEFGHIJKLMN
000050: PQRSTUVWXYZ[\]^
000060: `abcdefghijklmnop
000070: pqrstuvwxyz{|}~
000080: Ç.....
000090: .....
0000a0: !"#$%&'()*+,-.
0000b0: 0123456789:;<=>
0000c0: @ABCDEFGHIJKLMN
0000d0: PQRSTUVWXYZ[\]^
0000e0: E10      8EÄ99Ü9AÜ
0000f0:      84ä94ö81ü
pqrstuvwxyz{|}~

```

50. Codetable for reading BASIC files

The codetable consists of an address followed by the codes in hexadecimal. All other lines are comments.

The codetable is used for translation when reading a file from the disk. The opposite table for use when writing the file is constructed automatically, or may follow below in the textfile.

From version (006.xxx) the parity is included in the codetables itself and the buildin tables avoided if possible. This means the tables 1 and 2 (Danish and German buildin) now automatically uses the tables 045 and 049. To obtain this for BASIC codetables also an * mark just after SW-Tools in the first line reverses parity for all characters. The real character may be stated as a remark after the hexadecimal code or the hexadecimal code may itself be replaced with the character.

11.5.1. Codetable numbers

The codetables has been named using the country code numbers.

001	Danish UNIX Buildin
002	German UNIX Buildin
003	ODBC OemToAnsi Buildin
004	Parity plain 1:1 Buildin
010	ISO8850
011	DATAFLEX index
012	EBCDIC
013	ODBC OemToAnsi
044	English Basic parity
045	Danish Basic parity
046	Swedish Basic parity
049	German Basic parity
099	Sort multilingual

11.5.2. The SORT codetable

Sorting using RAPGEN or the SW-Tools ODBC driver uses the SORT codetable to sort language specific characters correctly (äü last) and accents together with the appropriate character.

11.6. System files

To use this the system file definitions must be installed. In this the system files for the Data-Dictionary, Subsystems, Reports and IQ programs are defined.

12. Database drivers

The Data-Dictionary supports a wide variety of database drivers and more will come. Some of these drivers are developed solely by SW-Tools and will in general be the fastest way to access the file system, but are restricted in updates especially concerning indexing. Others use the vendor's access routines in which case these must be present (bought separately). Such drivers may offer full update and build of files.

NOTE: SW-Tools drives supports searching forwards AND backwards in the index. For other drivers we can only offer the functionality built into the drivers which means if you use drivers such as CTRAS, IQ cannot display records in reverse order and you cannot go to previous record.

Even if updating files is possible you should note that this is done at your own risk. SW-Tools can take no responsibility of any damage to data due to any malfunction of any part of drivers or application programs. An update program should always be tested before taken into production.

12.1. SSV-Textfile driver

The SSV-Textfile driver is the internal file format for all SW-Tools products. This driver should always be present as the first driver.

The SSV text files has each record separated with Carriage return and each field with semi-colon which means ; is not allowed in any field. The files may be edited with any text-editor.

The SSV files are extremely fast for relatively small files and may be used both locally on a PC or placed on a server. Any sort of codetable may be given. Full update of the files is implemented but may only be done by one user at a time.

If you intend to create large files or update these in an multiuser environment you should choose a real database system.

12.1.1. Table name

The physical filename is generated using the standard path set for the driver and the table name. The extension of the file is normally .ssv, if not set. If no table name is given it uses the id, e.g. the table 'va' will access 'va.ssv'.

12.1.2. Field format

All fields are stored as textfields but may be defined as numeric. No pack options are used, byte numbers are not relevant.

12.1.3. Index description

The records are always physically sorted in the file according to the first (primary) index definition. If secondary indexes are used, the file is sorted whenever these are referred to.

12.2. ODBC Drivers

The ODBC drivers installed in Windows will be available in the Data-Dictionary. An installation example is given in the chapter 'Driver installation'.

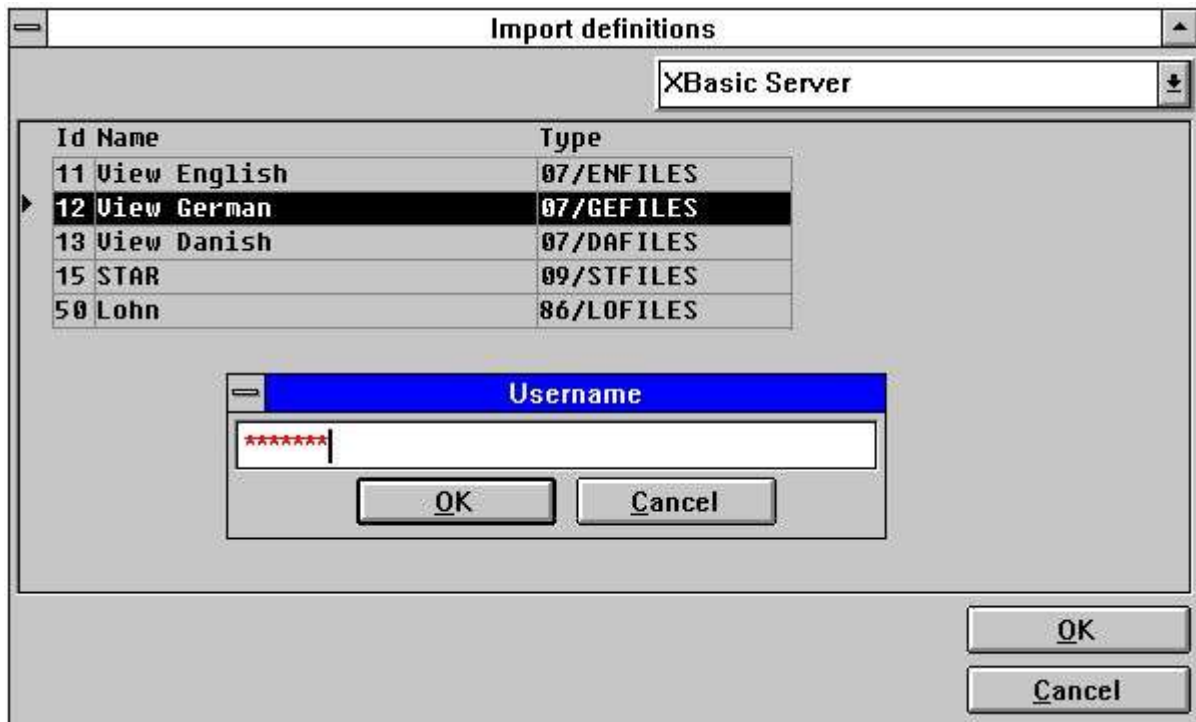
12.2.1. Codetable

The codetable should normally be ODBC (OemToAnsi) but may vary dependent on the application which created the tables.

12.3. BASIC drivers in general

The Basic drivers conform to the specification in the VIEW manuals. You can import file definitions directly from VIEW by a PUT followed by 'Get standard definitions', also the COMET master text files may be loaded directly.

With the function 'Import ODBC definitions' you can install a complete VIEW system from the server including any company dependent filenames. This function requires a password (BASIC) as the complete, and not a partial system, is imported.



51. Using Import ODBC definitions for a Basic driver

Some general parameters are used for all Basic drivers:

12.3.1. Field format

The BASIC file system stores fields values as BASIC variables, where the following field types may be used:

T1 1% = 1-word, max 7999, no decimals.

T2 2% = 2-word, max 6 digits

T3 3% = 3-word, max 10 digits

T4 4% = 4-word, max 14 digits

T5 5% = 5-word, max 18 digits, Unibasic only

The variable type is automatically calculated using the logical format. For example the field format 2, becomes a 1% integer, the format 9,2 a 3% etc.

12.3.2. Packed fields

Values stored in BASIC file system may be packed. Therefore the format might require a pack type set. The following pack types are supported:

- P** Value packed with CALL 60
- P1** Same as P
- P2** Value packed with CALL 60, no decimals
- P3** Field stored alphanumeric with decimals
- P4** Field stored alphanumeric without decimals
- P5** Stored as 1%, no decimals

12.3.3. Index descriptions

As described above indexes are all stored as text and may or may not be included in the data record. Constants in and packing of keys are supported.

12.4. BASIC Interfaces

The following lists the available BASIC interfaces:

Type	TRIO Interface	Other software needed	Write	Bits
XBasic-Unix	XBasic CTRAS (Library) CTRAS	CTRASX.DLL	Write Write	16 only
XBasic-NT	X-Basic			
Quattro	CTRAS Quattro (FTP)	CTRASQ.DLL SSQ	Write	16 only 32 recommended
X/Net	X/Net			
Netbasic	Netbasic CTRAS	CTRASQ.DLL (Netbasic)	Write	16 only
Unibasic	Unibasic			
Surfbasic	Unibasic			
CX-Basic	Unibasic			
OpenBasic	OpenBasic			

12.5. X/Basic driver

The X-BASIC driver can be used to access the X-BASIC file system on UNIX and PC systems. Update of files is restricted to non-index fields and records cannot be inserted or deleted.

Database interface		
Type	XBasic Server	OK
Internal Type	X-Basic	Cancel
Description	XBasic Server	
LU0 Path	/X.BASIC/0	Delete
User		
Password		Company
Server	200.0.0.3	
Connection type	TCP/IP	
Server password		
Code conversion	UNIX/DOS German	
Read only	Yes.	

52. Setting up an X/Basic driver

The database normally resides on a UNIX machine which then requires that the driver is setup to connection type Windows Sockets TCP/IP.

12.5.1. Table name

The physical filename is generated using the standard LU0 path set for the driver and the table name.

File ID:	<input type="text" value="jh Debtor/creditor master 1"/>	XBasic Server
Filename:	<input type="text" value="03/GF-03000abc"/>	Standard ID:
Realname:	<input type="text" value="/X.BASIC/0/3/GF-03000001"/>	<input type="text" value="GF-03000 2 GM"/>

53. X/Basic filename

12.5.2. Company number



54. Defining the standard company number

The company number is normally a 3-digit value 000-999. This value is used when the table name contains the reference 'abc' as 90/GF-03000abc.

12.6. CTRAS driver for Quattro or Sinix

CTRAS is a net driver system enabling you to connect to a Sinix or a Quattro system. CTRAS must be bought separately and is available in a 16 bit version only. CTRASX.DLL must be present on the system.

CTRAS allows full write access to the Basic file system. The descriptions for the functions mentioned below is to be found in the CTRAS manual. CTRAS is for the time being the only way to connect to a Quattro system. For Sinix systems you can use the server module delivered with TRIO.

Database interface	
Type	Ctr ^{as} Quattro
Internal Type	Ctr ^{as} Quattro/Sinix
Description	Ctr ^{as} Quattro
User	CMC
Password	*****
Server	Servername
Connection type	Normal
Server password	***
Code conversion	UNIX/DOS German
Read only	No.

55. Setting up an CTRAS driver

12.6.1. Servername and Server password

The servername states the net server used, the Server password is the password required to login on to this. These are passed as parameter 3 and 4 for the NF_CONNECT call.

12.6.2. User and User password

These are the user information on the host system, for a Quattro system the user could be CMC and the user password MANAGER. Passed as parameter 5 and 6 for the NF_CONNECT call.

12.6.3. Connection type

This must be NORMAL to invoke the CTRAS functions.



The image shows a dialog box titled "Ctras Quattro/Sinix". It has a blue title bar. The dialog contains three input fields and two buttons. The first field is labeled "Company" and contains the text "001". The second field is labeled "Quattro/Unix [1/2]" and contains the text "2". The third field is labeled "LU0 path" and contains the text "\$XBASIC.HOME". To the right of the input fields are two buttons: "OK" and "Cancel". A mouse cursor is pointing at the "OK" button.

56. CTRAS options

12.6.4. Quattro/Unix

Decides which host systems are used. Quattro is 1 and makes use of the TR_CONNECT functions in CTRAS, Sinix is 2 and uses the NF_CONNECT functions.

12.6.5. LU0 Path

For Sinix systems only a LU0 path must be given as /XBASIC/0 or if the server environment is set \$XBASIC.HOME. This is used in the NF_INIT call.

12.7. X/Basic Sinix C Interface Library driver

This way of running X/Basic does not require the CTRAS system but still it allows full write access for the Basic files.

The net communications are maintained with the TRIO unix server but the functions used to access the Basic files are the C Interface Library on Unix which is delivered free of charge as nfmlib.a together with the X/Basic system.

Database interface		
Type	X/Basic Library	OK
Internal Type	Ctras Quattro/Sinix	Cancel
Description	X/Basic Library	Delete
User		
Password		CTRAS-Setup
Server	200.0.0.3	
Connection type	TCP/IP	
Server password		
Code conversion	UNIX/DOS German	
Read only	No.	

57. Setting up an X/Basic Library driver

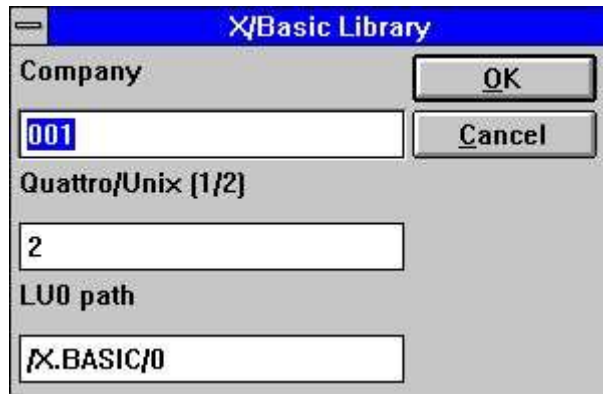
12.7.1. Connection type

This must be TCP/IP to invoke the TRIO Unix Server.

NOTE: On the SINIX system the server program (SWTUSOCK) must be started from a X/BASIC user to get the correct environment.

12.7.2. Options

The options for this driver are the same as for a CTRAS driver.



The image shows a dialog box titled "X/Basic Library". It contains the following fields and controls:

- Company:** A text field containing "001". To its right are "OK" and "Cancel" buttons.
- Quattro/Unix [1/2]:** A text field containing "2".
- LU0 path:** A text field containing "/X.BASIC/0".

58. X/Basic Library driver options

12.8. Quattro interface using 32 Bit FTP

An alternative to the rather slow CTRAS interface is now offered to the Quattro users which has a network connection and the SSQ FTP server available.

The interface uses FTP to pull over the complete files from the Quattro which are much faster than reading the single records whereafter a local read of the Quattro-format files are done. To allow long filenames we recommend the 32 bit version to be used as conflicts may occur if filenames are truncated to xxxxxxxx.xxx by the 16 bit version.

The files are transferred when opened by a report/IQ program or ODBC request and the frequency of update can be tuned for everytime, once a hour, once a day or like.

The screenshot shows a dialog box titled "Database interface" with the following fields and controls:

Type	Quattro using FTP	OK
Internal Type	Quattro	Cancel
Description	Quattro using FTP	
Home Directory	c:\dok\isa\	Delete
User	cms	
Password	***	Company
Server	200.0.0.9	
Connection type	ftp	Min 1440
Server password		
Code conversion	001 Danish UNIX Buildin	
Read only	No.	

59. Setting up Quattro using FTP

When activating the connection type as FTP input to the field MIN is activated. Here you may state the number of minutes between the FTP transfer of a file or in another way how old you will accept your data to be before new update.

Above is given 1440 = 24 * 60 minutes. Thus the files is updated once a day by firsttime usage.

If you leave MIN blank or set it to 0 files are refreshed each hour.

12.8.1. Working offline with the Quattro files

If you set MIN to -1 files are never transferred, you will continue to run on files once received from the Quattro.

If a file cannot be transferred due to connection is not available an error message is given. If an earlier transferred file is available the program will use data from this without new transfer. You may transfer files manually to the PC from a quattro system also using the FTP (see this) or tape transfer. The Quattro interface are working with exact binary copies of the Quattro files including the header block.

12.8.2. File definitions

No direct import of the Quattro file definitions are available, these must be loaded otherway.

12.8.3. XBasic and other interfaces using 32 Bit FTP

In practice you may use any of the builtin interfaces to transfer files using FTP and then running with local files. For example XBASIC on a RM system may be set up as:

The screenshot shows a dialog box titled "Database interface" with the following fields and controls:

Type	XBasic-FTP	OK
Internal Type	X-Basic	Cancel
Description	XBasic-FTP	
LU0 Path	/DATA/isa	Delete
User	root	
Password	*****	Company
Server	200.0.0.2	
Connection type	ftp	Min 60
Server password		
Code conversion	002 German UNIX Buildin	
Read only	No.	

60. Setting up XBasic using FTP

However one path only is available which means the local path will become the same as the path on the server. The local path will be created if not present.

By the way you should note that the XBasic interface, also in CTRAS-Library write mode, may be used on an UNIX system even if XBasic itself is not installed.

12.9. XNet driver

XNet is a Basic system residing on the PC or a network drive on the PC. No special server is required apart from the already installed net software (NOVELL or Windows).

Update of files is restricted to non-index fields and records cannot be inserted or deleted.

The screenshot shows a dialog box titled "Database interface" with a blue header bar. It contains several fields and buttons:

Type	XNet	OK
Internal Type	X/Net	Cancel
Description	XNet	
X/Net Directory	d:\xnet	Delete
User		
Password		Company
Server		
Connection type	Normal	
Server password		
Code conversion	UNIX/DOS German	
Read only	Yes.	

61. Setting up a XNet driver

12.9.1. XNet filenames

On Xnet the Basic filename is transferred into a DOS filename using an INDEX file residing on the XNet main directory. The filenames are just internal numbers with no relation to the original name.

File ID:	dk Debtor file	XNet
Filename:	90/GF-03000abc	Standard ID:
Realname:	d:\xnet\lu3\x1002201.003	GF-030001 2 GM

62. XNet filename

12.10. Netbasic driver

Netbasic is a Basic system residing on the PC or a network drive on the PC. No special server is required apart from the already installed net software.

Update of files is restricted to non-index fields and records cannot be inserted or deleted.

Database interface		
Type	Netbasic	OK
Internal Type	Netbasic	Cancel
Description	Netbasic	
NetBasic directory	d:\netbasic\	Delete
User	NBENV10.CFG	
Password		Company
Server		
Connection type	Normal	
Server password		
Code conversion	UNIX/DOS German	
Read only	Yes.	

63. Setting up a Netbasic driver

12.10.1. User

This field may state the name of the LU configuration file if not NBENV3.CFG.

12.10.2. Netbasic filenames

Unfortunately the structure of these is dependent of the version of Netbasic you are using. You will have to check the compatibility of your actual version.

Old versions did use a NBENV3.CFG and FILELIST.CFG with the lu's and filenames. The version here supported (Netbasic 4.25) uses NBENV10.CFG for the lu's and an internal formula to transform the Basic into a non-readable Dos filename.

File ID:	1a Article mainfile	Netbasic
Filename:	03/LF-06000001	Standard ID:
Realname:	D:/NETBASIC/LU3/ÜÆIµ_B".+ _	LF-06000001

64. Netbasic filename

12.10.3. NETBASIC using CTRAS functions

For NETBASIC a CTRAS-like DLL may be supplied from your Netbasic supplier which opens the possibility of both reading and writing to Netbasic files.

The DLL must be installed if you are having Netbasic version 4.29 or higher, with version 4.28 or lower you may use the SW-Tools buildin Netbasic interface.

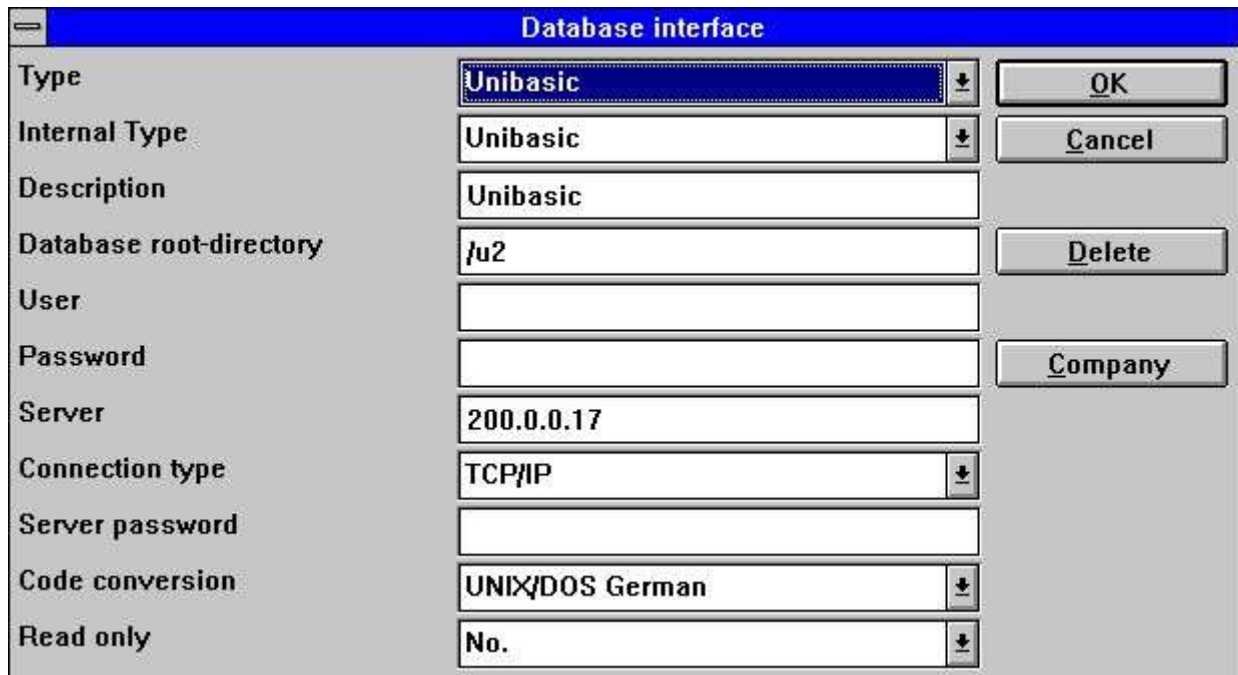
When using the DLL you should follow the installation guide for CTRAS interfaces, as well as you should have a CTRAS license for TRIO.

Unlike the Quattro/XBasic CTRAS interface reading backwards in files are implemented for Netbasic.

12.11. Unibasic / Surfbasic / CX-Basic / Open-Basic driver

The structure of Unibasic, Surfbasic and CX-Basic files are internally the same and the driver used and setup procedure for these are identical.

Update of files is restricted to non-index fields and records cannot be inserted or deleted.



The screenshot shows a dialog box titled "Database interface" with the following fields and buttons:

Type	Unibasic	OK
Internal Type	Unibasic	Cancel
Description	Unibasic	
Database root-directory	/u2	Delete
User		
Password		Company
Server	200.0.0.17	
Connection type	TCP/IP	
Server password		
Code conversion	UNIX/DOS German	
Read only	No.	

65. Setting up an Unibasic driver

12.11.1. Unibasic files

Index files are splitted into two parts, UPPERCASE filename is the index and lowercase the data part.

The header of the file indicates if the old or new index system is used. Both systems are supported.

However if you need a testsystem on a PC you should copy the index part to xxxxxxxx.dat and the data part to xxxxxxxx.idx as upper/lowercase filenames are not supported.

12.12. NAVISION Financials

The 32 bit version of TRIO and the ODBC driver now interfaces the Navision Financials package by use of the C-Front module which must be present on the system. Both read and write are implemented, C/FRONT version 1.2 or higher is supported.

12.12.1. Install Navision Financials and C/Front

Install Navision Financials and C/Front for example in directory e:\fin and make sure that the PATH environment variable includes the directory e:\fin. If you change the PATH variable restart the Windows system.

12.12.2. Install and configure the Navision Financials interface

Follow the description for installing a ODBC driver interface in TRIO. The interface for Navision is named 'Navision Financials'. By the function 'Database-interface' you may select the installed 'Navision Financials' where the following setup will correspond to the described sample installation.

The screenshot shows a dialog box titled "Database interface" with the following fields and values:

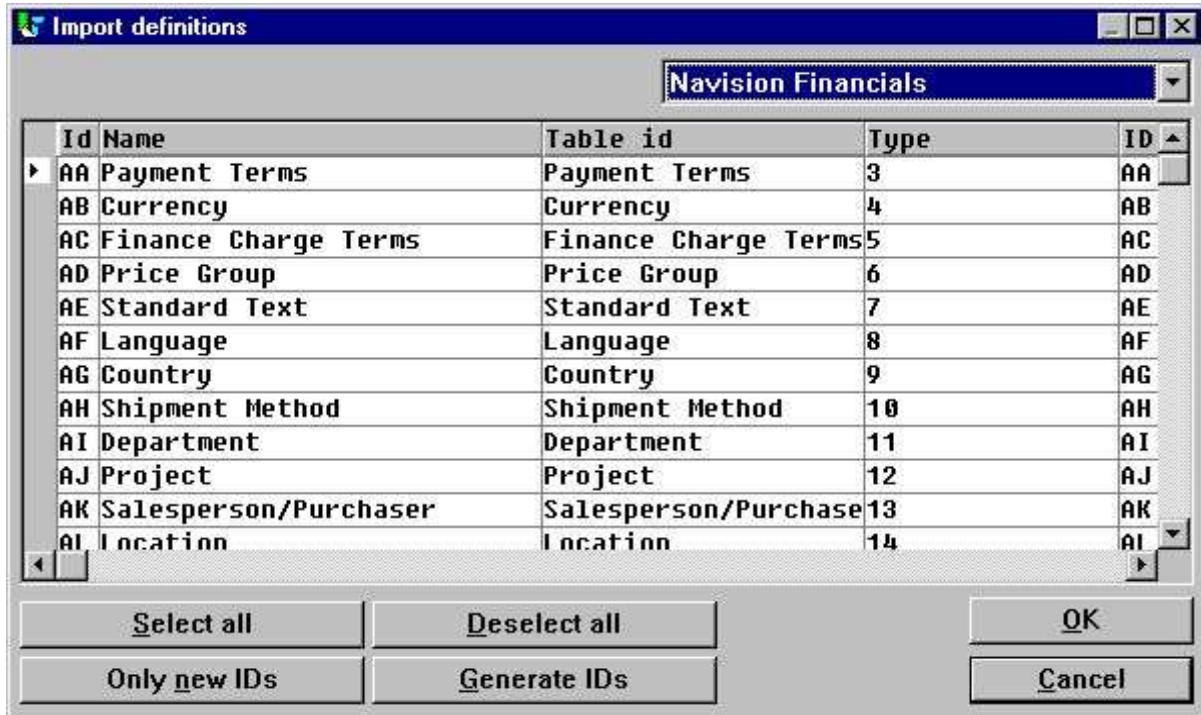
Type	Navision Financials	OK
Internal Type	Navision Financials(19)	Cancel
Description	Navision Financials	
Home Directory	e:\fin\database.fdb	Delete
User		
Password		Company
Server		
Connection type	Normal	
Server password		
Code conversion	None	
Read only	No.	

66. Sample setup of Navision Financials interface

If you use C/Front with TCP/NETB for server connection the field Server may be used to enter 'server,type' where server is the server name and type is tcp or netb.

12.12.3. Import of table definitions

From the menu 'File' - 'Import ODBC definitions...' you may select interface 'Navision Financials' in order to import all or some of the table definitions.



67. Import of all or some of the Navision table definitions

12.13. CONCORDE C5/XAL

Concorde C5/XAL interface is available as 16 or 32 bits as readonly for TRIO and the ODBC driver.

12.13.1. Exporting the Concorde file definitions

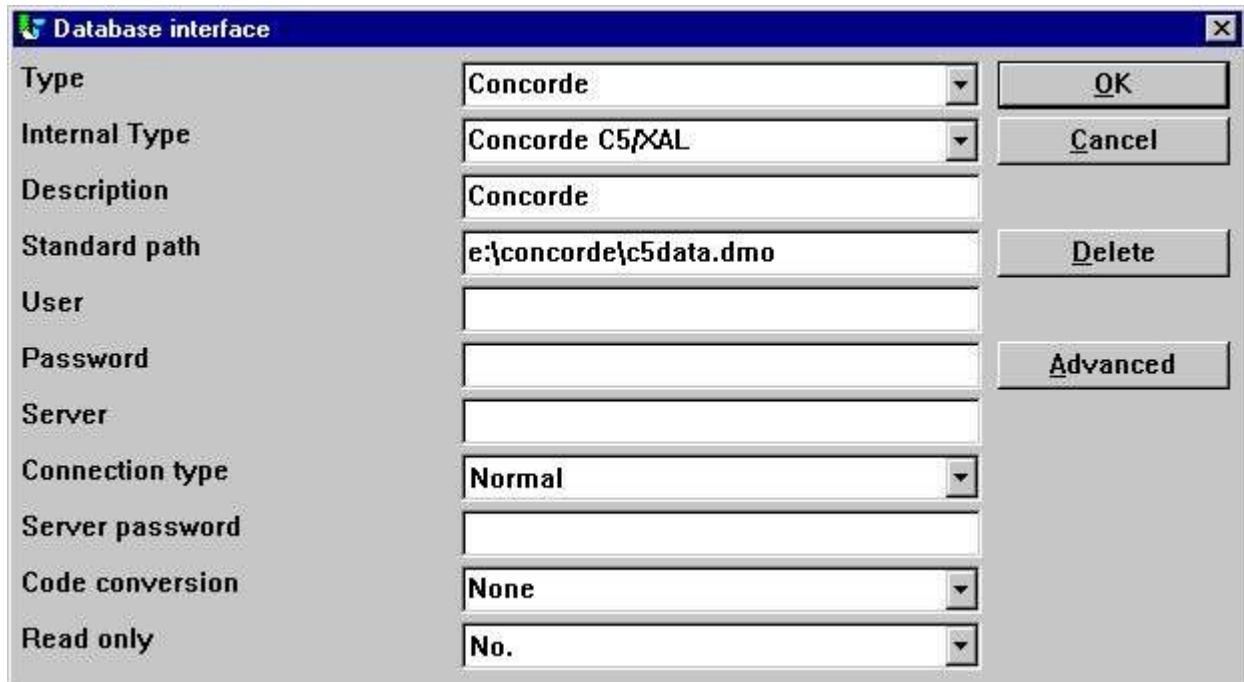
If Concorde is installed for example in directory e:\concorde you should first export all file definitions into a .DBD file.

The export may be done by login as 'supervisor' and select the menu 'General' - 'Amendments' - 'Development menu'. From here you may select 'Export' - 'DBD' - 'All + Enum'. The exported file is then saved in the same directory as the Concorde installation, e.g. e:\concorde.

12.13.2. Install and configure interface to Concorde

Follow the description for installing a ODBC driver interface in TRIO. The interface for Concorde is named 'Concorde C4' or 'Concorde C5/XAL'.

By the function 'Database-interface' you may select the installed 'Concorde' where the following setup will correspond to the described sample installation.



The screenshot shows a dialog box titled "Database interface" with a close button (X) in the top right corner. The dialog contains several fields and buttons:

Type	Concorde	OK
Internal Type	Concorde C5/XAL	Cancel
Description	Concorde	
Standard path	e:\concorde\c5data.dmo	Delete
User		
Password		Advanced
Server		
Connection type	Normal	
Server password		
Code conversion	None	
Read only	No.	

68. Sample setup of Concorde interface

12.13.3. Import of table definitions

From the menu 'File' - 'Import ODBC definitions...' you may select interface 'Concorde' in order to import all of the table definitions.

A user name is required in order to import definitions. The user name is 'BASIC'.

When the user name has been entered the import will be of all tables, e.g. no selection can be made. If required you may delete one or more of the imported definitions afterwards.

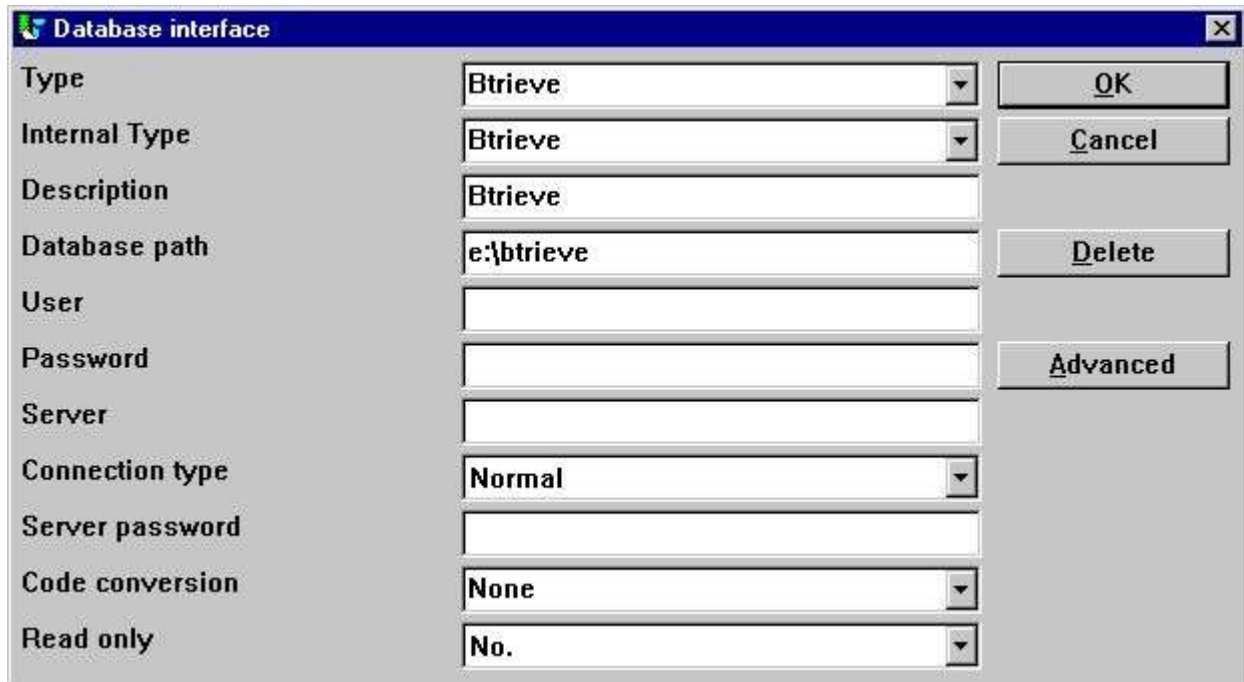
12.14. Btrieve version 5.10 and 6.15

Btrieve files can be accessed directly without use of ODBC. This will give a much better performance when working with TRIO.

12.14.1. Install and configure interface to Btrieve

Follow the description for installing a ODBC driver interface in TRIO. The interface for Btrieve is named 'Btrieve'.

By the function 'Database-interface' you may select the installed 'Btrieve' where the following setup will correspond to access of Btrieve files located in the directory 'e:\btrieve'.



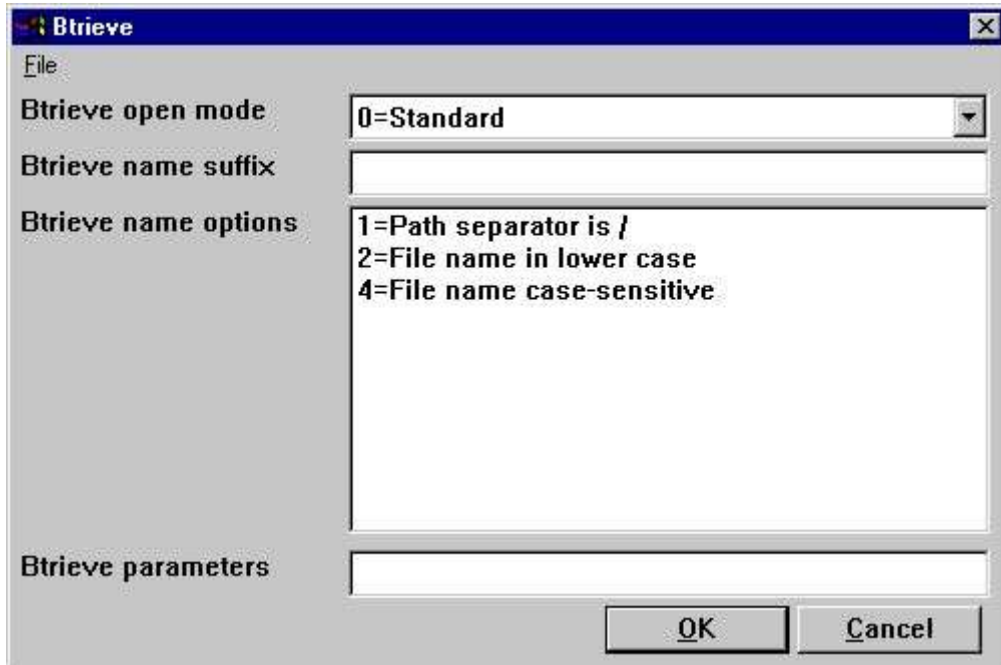
The screenshot shows a dialog box titled "Database interface" with a close button (X) in the top right corner. The dialog contains several fields and buttons:

Type	Btrieve	OK
Internal Type	Btrieve	Cancel
Description	Btrieve	
Database path	e:\btrieve	Delete
User		
Password		Advanced
Server		
Connection type	Normal	
Server password		
Code conversion	None	
Read only	No.	

69. Sample setup of Btrieve interface

12.14.1.1. Advanced options

The Btrieve interface allows you to control how files are to be opened and how the file name should be generated.



70. Advanced settings on Btrieve interface

12.14.1.1.1. Open mode

The open mode may be one of the following:

- **Standard**
- **Exclusive**
- **Single Engine File Sharing (Version 6.15 only)**
- **Multi Engine File Sharing (Version 6.15 only)**

By standard TRIO will open a Btrieve file in read-only mode unless it is a file which is to be updated on a report or in a DATAMASTER program.

When the open mode is Exclusive, no other program can gain access to the file.

The Single and Multi Engine File Sharing modes are for Btrieve version 6.15 only. Please refer to the Btrieve documentation for a detailed description of the openmodes.

12.14.1.1.2. Name suffix

The name suffix of a Btrieve file was always set to .DAT in TRIO version \leq 006.008. However, Btrieve files may be without any suffix or a different one than .DAT. Therefore, it is a field you may fill yourself when installing the interface for use in TRIO.

12.14.1.1.3. Name options

These options controls how the file name of a Btrieve file is generated by TRIO. The naming options for a Btrieve may override the normal generation of the actual file location. As an example, assuming that the interface has been setup with database path

c:/btrieve/database

and a file is defined with name

customer

and the suffix is set as

dat

the actual location of the file is

c:/btrieve/database/customer.dat

Because Btrieve does NOT allow the use of / in the path the default behaviour is to replace all occurrences of / with \ giving a location as

c:\btrieve\database\customer.dat

Should this however change in newer Btrieve versions or only be dependent on the operating system used, it can be changed here.

12.14.1.1.4. Parameters

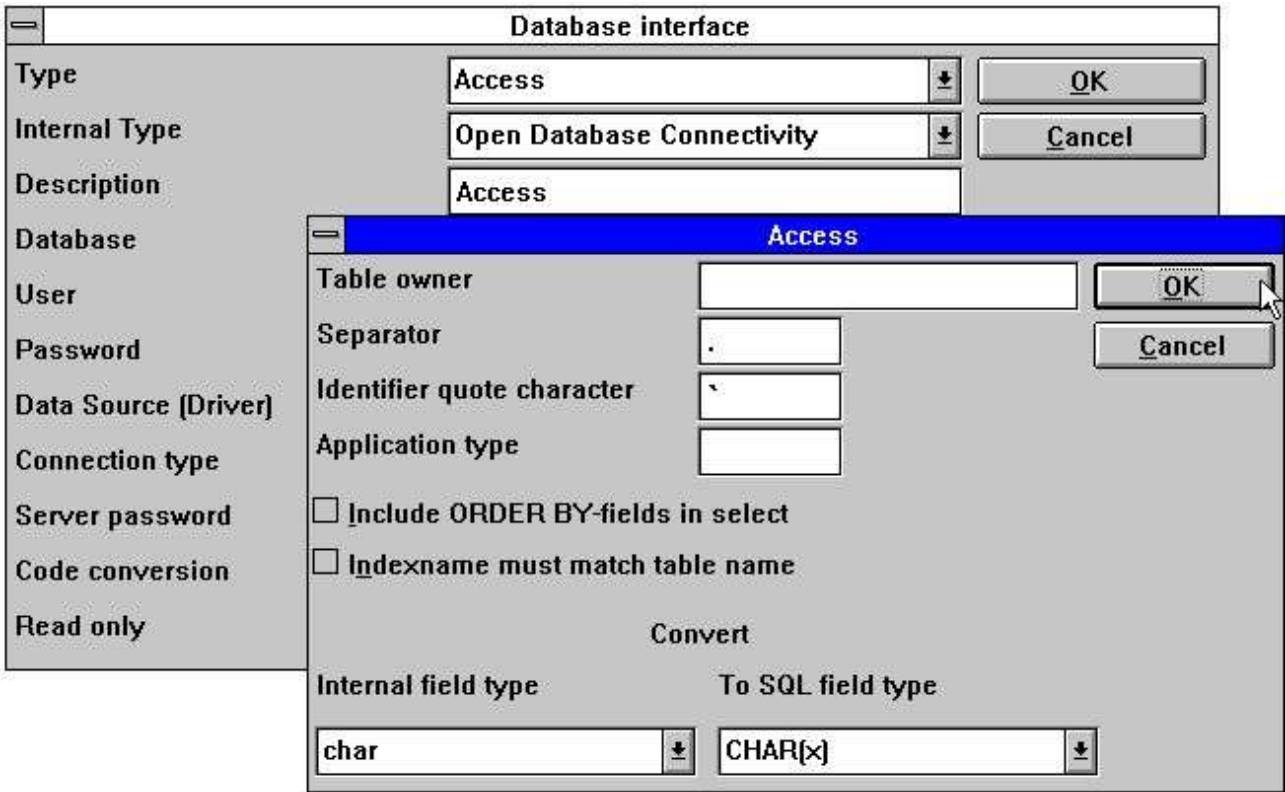
When Btrieve 5.10 is activated it is required to pass some initial parameters to the Btrieve system. The parameters are by default:

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

Please refer to the Btrieve 5.10 manual for a detailed description of the possible parameters and the meaning hereof. The parameter string has no effect when using Btrieve 6.15 or newer.

12.15. ODBC driver setup

As some ODBC drivers are requiring special information which cannot be retrieved or trusted from the driver itself the ADVANCED setup for ODBC drivers has been extended with the following:



71. Setting up an ODBC driver

You should refer to the actual documentation for your driver before changing these fields.

12.16. ODBC drivers interface - Application type

An application type option can be used to control how TRIO should integrate to an ODBC driver as there may be differences in the SQL language / options dependent on the driver supplier. Note that more types may be set at the same time by adding the values.

12.16.1. Type 1 - Support of Informix Database table locks

The normal use of an ODBC data source for Informix will lock any used table when connected to it. This will for example cause a problem if a user is working with IQ or RAPGEN to query or print information from the Informix database at the same time when a user wants to work with the table from another application.

It is described in the Informix database documentation that you have to execute a SQL statement 'set isolation to dirty read' to avoid the problem.

12.16.2. Type 4 - Always perform ORDER BY

Because not all ODBC drivers support the standard given by Microsoft, you may experience SQL errors when trying to print reports with access to a table using the ODBC driver. The error will state a missing ORDER BY ... for the table.

A report that selects the fields from a customer table, where customer is the primary key will generate

```
SELECT customer,name,address FROM customer_table
```

If type 4 is set it will generate

```
SELECT customer,name,address FROM customer_table ORDER BY customer
```

12.16.3. Type 4096 - Alpha fields may NOT be NULL

Because some ODBC drivers / SQL Databases does NOT support alpha numeric fields with no content, e.g. "" it is necessary to inform TRIO that all alpha fields must contain some value. This type will automatically use the field pack type 1048 on all fields, which will set the content of an empty alpha field to one space.

12.16.4. Type 8192 - ODBC Drivers with support of one connection per table only

Because some ODBC drivers / SQL Databases requires a unique connection for each table, where as TRIO tries to optimise table access by doing the connection to the database only once it can be necessary to set this type.

If this type is set, TRIO will create a connection to the database for each table used. Please note, if the ODBC data source requires the user to enter a user name / password to connect to the database, it will be required to do so for each table used. To prevent this, the user name and password must be setup one for the database interface in TRIO.

12.17. ISAM database interface - Application type

The application type may be used for ISAM also for company control:

12.17.1. Type 2 - Support of SAMSON system

The SYSTEM named SAMSON, is a financial system for DOS/Windows and UNIX, using C-ISAM as Database. In this system, all files are created with a company number as the first key part always. TRIO supports the use of company numbers on interfaces and sub systems, but normally only in connection with the file name.

13. Other drivers / Notes on some application packages

The following other drivers are available on request

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

13.1. GLOBAL 3000 Speedbase and Global 2000 ISAM/DMAM

A separate installation manual is available on request.

13.2. ALX database

The COMET/ALX solution provides an Informix database that can be accessed, with an installed ODBC driver.

In order to get access from the Data-Dictionary perform the following tasks:

- 1. Install the required driver, ODBC driver for Informix.**
- 2. Set the driver parameter user, for example alxdemo**
- 3. Set the driver parameter password, for example alxdemo**
- 4. Set the server name, for example dbm216**
- 5. Import ODBC tables**

When tables have been imported each table can access some special company. The company has to be entered in the table name. For example, if the company 999 is required for the table ac2010:

"c999".ac2010

13.3. SSV database interface

The SSV interface for Semicolon Separated Textfiles has from TRIO (007.001) been completely reworked and optimised for speed improvement and compatibility on keystructure. Compressed files has been implemented for disk space saving also.

You do normally not have to set up any options for the SSV interface but the following ADVANCED options will be available:

The screenshot shows a dialog box titled "SSV Textfile" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog contains several input fields and dropdown menus for advanced options. The options are:

- File
- Write sortindex
- Headerlength
- First data record
- Make indexfile > records
- Write CRLF
- Indexfile naming (dropdown menu showing "0=Filepath")
- Duplicates >= index
- Read CRLF
- Compression used (dropdown menu showing "0=None")
- Initial codetable

At the bottom of the dialog are two buttons: "OK" and "Cancel".

72. The ADVANCED options with the SSV driver

Note that when you move the mouse cursor over the field leading text you will get floating online help displayed.

13.3.1. Write sortindex

Determines sorting by write of the file.

0 = No sorting, recordnumbers (lines) are kept

>0 = Sort the file according to this index definition

13.3.2. Headerlength

Used by the SSV interface only, sets the first valid data byte of the file.

13.3.3. First data record

First data record used in the file, records below this number are ignored.

First real data position is calculated as: $\text{Headerlength} + \text{First data record} * \text{Recordlength}$

In case of SSV files containing lines of text the first data record indicates the first line to be used. The default when blank is 1 causing a headerline to be reserved. Set to 0 if you want the first line to be included. The field may be given as:

512 Decimal

0x200 Hexadecimal

***8,B16P1006**

where the last line marked with an leading * states that a field with the format 8, should be read from the file exact byte position 16 and unpacked using packtype 1006 used as value here.

13.3.4. Make indexfile > records

Huge SSV files may require time sorting each time the file is opened and an index is used. This parameter defines that for files larger than X records an external indexfile should be kept and used whenever possible. The default is to keep indexfiles for files larger than 10000 records.

Such files may be placed on the TMP directory or together with the file itself depending on the naming parameter.

13.3.5. Write CRLF

A SSV file will normally be written using the same CRLF characters as original read from the file. On a new file LF = 0x0A is used. You may specify a fixed crlf for write hexadecimal here as:

0x0a = LF only
0x0d = CR only
0x0a0d = LF+CR
0x0d0a = CR+LF

13.3.6. Indexfile naming

If an indexfile is build for large SSV file, the filepath is determined based on this parameter:

0 = filepath/filename.Xnn

1 = TMP/filename.Xnn

13.3.7. Duplicates \geq index

Duplicates will normally not be allowed in an index.

If you define a matchcode index you will normally have to separate equal keys yourself which may be done by adding recordnumber as NP to the key definition.

However you have the possibility also to define that all index above the here given number will allow duplicate keys. When a SSV file is read all index will always allow duplicates anyway as the build of the file may be done in many ways.

13.3.8. Read CRLF

When reading a SSV file the CRLF sequence will normally be automatically detected and used also by potential write. However you may fix the crlf as for the write crlf, see above.

13.3.9. Compression used

By use of 1 the file may be compressed using the LZ standard in Windows with the COMPRESS / EXPAND programs saving space.

When you set 1 a compressed file will be autodetected and read just as any other file, however you will have to manually compress the files as writing will always be done uncompressed.

0 = No compression favours the fastest read speed for the file.

1 = May be compressed

13.3.10. Initial codetable

The normal codetable specified for the driver is used by converting individual alphanumeric fields from the file.

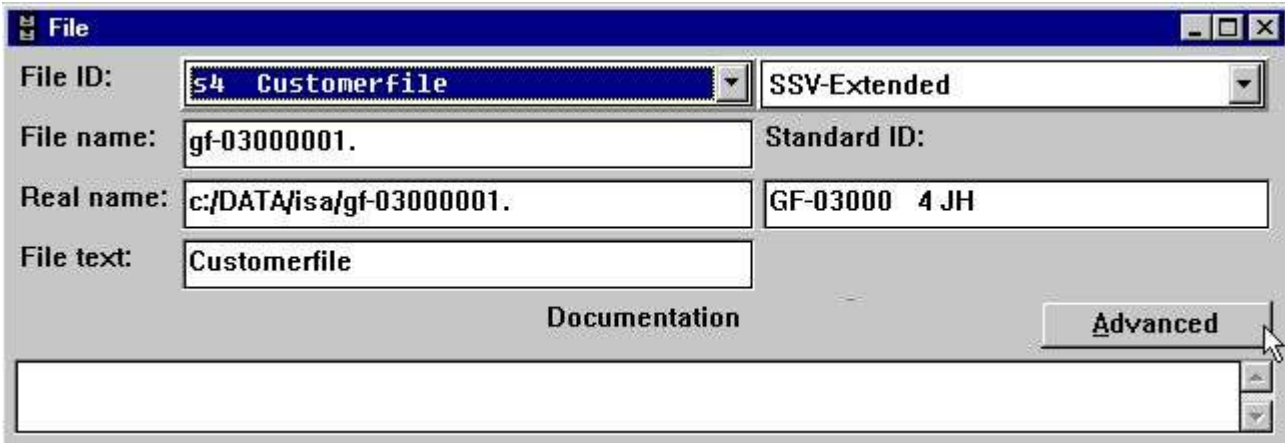
By giving a codetable number here you can convert the complete file directly after read or directly before write meaning that the field separator characters and linefeeds may be converted also.

13.4. Extended SSV interface

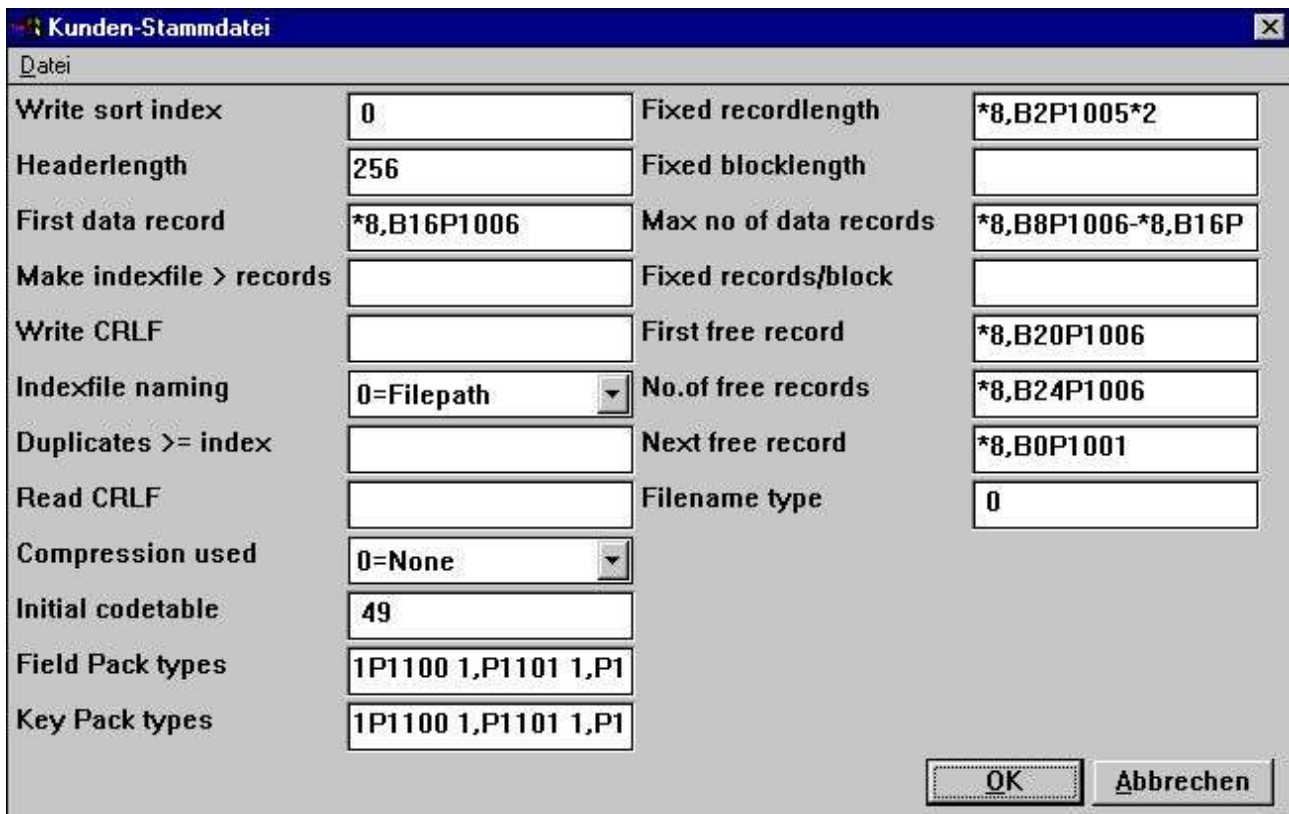
Extended SSV parameters for each file are now available whereby not only textfiles but many plain files fits into this interface.

Even complex filestructures as fixed recordlength X-Basic indexed files may be defined naturally just for direct access only skipping the index part. However setting up such driver requires you have excessive knowledge of the file system details.

The ADVANCED options may be set by the driver installation or for each file also as an PARAMETER button is added to the main file information screen:



73. The PARAMETER button for single files.



74. The ADVANCED PARAMETER for the extended SSV driver defining a X/Net file

13.4.1. Fixed recordlength

Filling this field changes a SSV file from variable to fixed recordlength, enabling the extended SSV interface to access many different filetypes. The field may be:

***8,B2P1005*2**

stating that a field with the format 8, should be read from the file exact byte position 2 and unpacked using packtype 1005 and multiplied by 2.

13.4.2. Fixed blocklength

Together with a fixed recordlength a blocklength may be given fitting a certain number of records into one block and possible leaving a gap between the blocks. The blocklength may be given directly or as an number of records/block. The field may be:

***8,B2P1005*2**

stating that a field with the format 8, should be read from the file exact byte position 2 and unpacked using packtype 1005 and multiplied by 2.

13.4.2.1. Max no. of data records

A SSV file may be restricted to use a maximum number of records within a larger database or diskarea. The field may be:

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

stating that two fields with the format 8, should be read from the file exact byte position 8 and 16 and unpacked using packtype 1006 and subtracted from eachother.

13.4.3. Fixed records/block

Together with a fixed recordlength a blocklength may be given fitting a certain number of records into one block and possible leaving a gap between the blocks. The blocklength may be given directly or as an number of records/block.

13.4.4. First free record

A first free data record may be given to the SSV interface in order to skip over reading all free data records by following the free record chain. The field may be:

***8,B20P1006**

stating that a field with the format 8, should be read from the file exact byte position 20 and unpacked using packtype 1006.

13.4.5. No.of free records

A number of free data records may be given to the SSV interface in order to skip over reading all free data records by following the free record chain. The field may be:

***8,B24P1006**

stating that a field with the format 8, should be read from the file exact byte position 24 and unpacked using packtype 1006 and used as value here.

13.4.6. Next free record

A free data record chain may be defined where each free data record points to the next in order to skip over reading all free data records by following the free record chain. The field may be:

***8,BOP1001**

stating that a field with the format 8, should be read from the actual record byte position 0 and unpacked using packtype 1001 used as pointer to the next free record.

13.4.7. Filename type

Reserved for future use, leave this field as 0.

Determines which driver interface should generate the filename and possible other parameters for use by the SSV.

13.4.8. Field Pack types

May be used by fixedlength files only, a normal SSV file always uses SSV packing.

The packtypes may be given directly on the fields as e.g. P1013, or the default may be given here for each fieldtype separated by blank.

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

gives the normal BASIC packing of Alpha, 1 word, 2word, ... , Call 60 etc.

The first 6 being for fieldtypes 0-5 followed by P0-9

13.4.9. Key Pack types

As fields packtypes but used when a field is used in a key, se above.

A normal SSV file always uses LDCHAR/CALL60 when a field is places in a key.

Figure list

1. Licence screen.....	9
2. Data-Dictionary	10
3. Accessing the functions using menus or the toolbar	11
4. Other buttons on the toolbar	12
5. General menus and buttons	12
6. Functions.....	13
7. Selecting the driver installation function	14
8. Installed drivers	15
9. Installing new drivers.....	15
10. Selecting the driver interface function	16
11. Changing driver parameters	17
12. Importing ODBC definitions	29
13. Standard definitions from the BASIC version of SW-Tools File definitions	36
14. Putting definitions to a textfile	43
15. Id of new table	46
16. Setting the driver for a table	49
17. Database relations enquiry.....	55
18. Deleting tables	56
19. Record overview	57
20. Reserved words	58
21. Editing fields	59
22. Inserting and deleting fields	59
23. The VIEW menu.....	88
24. SQL-Names and SQL-Types.....	91
25. Byte, C-Byte, C-Variable and Basic variable.....	93
26. Record contents for table	96
27. Hexadecimal dump of a table	99
28. Help overview.....	102
29. Confirmation of DATAMASTER file copy	106
30. The Edit menu	108
31. Editing indexes	109
32. Inserting and deleting index lines.....	109
33. AF-05000/020 Sales order lines index and connections.....	115
34. Example of pseudoindex and recordnumber definition	132
35. Access orders table with SQL names	134
36. Floating on-line field help.....	146
37. Editing field help description.....	147
38. Editing field check.....	148
39. Printing documentation.....	151
40. Selecting printer	152
41. Documentation options	153
42. Example of table Documentation.....	154
43. Example of table of contents	154
44. Preferences	155
45. The subsystem menu	178
46. Opening subsystems	179
47. Defining a subsystem	180
48. A company selector.....	191
49. Company filenames.....	196
50. Codetable for reading BASIC files.....	199
51. Using Import ODBC definitions for a Basic driver.....	210
52. Setting up an X/Basic driver	215

53. X/Basic filename	216
54. Defining the standard company number.....	217
55. Setting up an CTRAS driver	218
56. CTRAS options.....	221
57. Setting up an X/Basic Library driver	224
58. X/Basic Library driver options	226
59. Setting up Quattro using FTP.....	227
60. Setting up XBasic using FTP	230
61. Setting up a XNet driver	231
62. XNet filename.....	232
63. Setting up a Netbasic driver	233
64. Netbasic filename	235
65. Setting up an Unibasic driver.....	237
66. Sample setup of Navision Financials interface	241
67. Import of all or some of the Navision table definitions	242
68. Sample setup of Concorde interface	245
69. Sample setup of Btrieve interface	248
70. Advanced settings on Btrieve interface	249
71. Setting up an ODBC driver	254
72. The ADVANCED options with the SSV driver	265
73. The PARAMETER button for single files.....	276
74. The ADVANCED PARAMETER for the extended SSV driver defining a X/Net file.....	277

Index

A

Access 16;24;44;76;79;129;257;283
 ALX 259
 Ascii 96

B

Based 46
 BASIC
 31;46;52;82;102;121;125;165;166;192;
 193;194;205;206;207;209;210;220;241;
 281;283

BASIS 182;185;188
 Binary 79
 Bit 222;225
 BTRIEVE 45
 Byte 84;88;283

C

C4 240
 C5 238;240
 CALL 79;113;207
 C-byte 88
 C-ISAM 3;44;256
 Codetable 194;195;204;283
 CODnnn 194
 COMET ... 39;49;110;111;144;161;205;259
 COMPANY 185;191
 Compress 128
 Concorde..... 78;238;239;240;241;284
 C-Structure..... 88
 CTRAS
 198;209;213;216;217;219;221;222;225;
 231;284
 CTRASX 209;213
 C-variable..... 78;89
 C-Variable 88;89;283
 CX-Basic 79;209;232

D

Database
 11;12;16;50;130;153;180;182;188;198;
 236;240;243;251;256;283
 Databaselock 182
 Datadict 185;187
 Data-Dictionary
 1;3;4;5;6;7;8;10;19;24;31;59;141;152;
 197;198;203;259;283
 Dataflex 78;257
 DATAMASTER
 72;73;74;101;102;121;141;143;145;171
 ;245;283
 Datasyst..... 185;187
 DATE 75;78;79
 Deleting 22;51;283

Descending 128;132
 Digital 257
 Directory 107;122
 DLL..... 209;213;231
 Dmsystem 185;189
 Documentation..... 3;48;148;149;283
 DOS..... 12;19;163;193;227;256
 Driver 9;10;25;32;44;78;203

E

EDIT 86;103;142
 ENG 194
 Excel..... 24;44

F

Field
 54;79;85;141;143;144;201;206;207;281
 FILES 152
 Fixed 136;273;274;276
 Float 78
 Format 57

G

GET 38

H

HELP 142
 Hexadecimal 94;263;283
 HOSTS 18

I

Id 26;33;41;43;283
 Import 205;237;241;259;283;284
 Index
 72;103;104;109;115;121;127;130;202;2
 08;233
 INDEX 121;137;227
 Indexname 105
 Informix 251;259
 INI 151;172
 Inserting 54;104;283
 Installation 10
 Integer..... 78
 Interface 209;219
 IP-address..... 18
 IQ
 54;121;124;141;143;159;160;169;172;1
 74;184;197;198;222;251
 Isam 78;79;109
 ISAM..... 116;255;257;258
 ISO8850..... 19;195

K

Key..... 108;120;122;128;282
 KNIR 120;122
 KOM 191

L

Language 19;155
 Layout85
 Leading 58;156
 Length79
 Library209;219;221;225;284
 Licence.....4;283
 LOAD 37;39;49
 LOGICAL 111;121;137;139
 Long79
 LU 192;211;218;229

M

MANAGER.....215
 Microsoft 123;252

N

NBENV10.....230
 NBENV3 229;230
 Netbasic209;228;230;231;284
 NF_CONNECT.....214;215;217
 NF_INIT 218
 NOVELL.....226
 NP 116;267
 Numeric78

O

ODBC
 .3;16;17;18;19;21;23;24;28;44;45;53;7
 5;76;78;86;87;102;109;129;145;195;19
 6;203;204;205;222;234;236;237;238;24
 0;241;242;243;249;250;251;252;253;25
 4;259;283;284

OemToAnsi 19;195;204
 Open 232;245
 ORDER 109;131;162;252

P

Pack 281;282
 Packed 77;207
 Packing 77;113
 Packtype 102
 Parameter80
 Password.....177
 Path.....218
 PATH 235
 PC 126;151;180;199;210;223;226;228;233
 Preferences 150;172;179;283
 Printer 147
 PUT 31;37;38;49;205

Q

Qnnn75;87
 Quattro
 209;213;215;217;222;223;224;231;284

R

RAP 151;172
 RAPGEN141;196;251

READ 121;136;137;138;139;140;144
 Record 52;91;93;114;122;283
 Recordlength263
 Right.....69;78;79

S

Secondary80;81
 SELECT 131;252
 Separators.....158
 Server..... 18;192;214;220;236
 Servername214
 SID..... 37;39;49
 Sign.....156
 SINIX.....220
 Sort 195;261
 Spool154

SQL

 75;76;86;87;100;109;129;130;145;250;
 251;252;253;254;283
 SQL-Name 86;130;283
 SQL-types 87
 SSD152
 SSV
 10;126;152;182;185;191;199;260;262;2
 63;264;265;266;267;268;271;272;273;2
 75;277;278;280;281;282;284

Sticky185
 Subsystem188
 Surfbasic 209;232
 SY43;71
 SYSTEM..... 172;256

T

Table-id..... 28
 TCP..... 18;19;210;220;236
 TEMP154
 TIME 75;79;126
 TIMESTAMP 75
 TMP154;264;266
 TR_CONNECT217
 Trailing.....156
 TRIO

 3;24;54;55;70;72;90;105;126;141;143;
 145;161;166;209;213;219;220;231;234;
 236;238;240;242;243;245;246;247;250;
 253;254;256;260

U

UNIBASIC 44
 UNIX.....12;18;19;195;210;225;256
 UPDATE..... 76
 UPPERCASE233

V

VIEW 83;130;205;283

W

Windows
18;19;147;151;172;203;210;226;235;25
6;269

WW43

X

XAL 238;240

X-BASIC 44;210
XNet 182;226;227;284

Y

Year. 57;80;157;161;162;164;165;166;167

Z

Zero 58;117