SWT
SOFTWARE TOOLS

English User Manual

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

DATAMASTER User manual

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

Contents

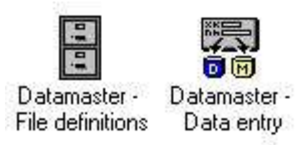
1. Introduction

SW-Tools DATAMASTER is a tool which allows the user - without any previous experience in programming - to define tables/files, maintain them and expand their contents and size.
A table defined with DATAMASTER is automatically installed into our other products:

- RAPGEN can be used to produce printouts / batch update programs and
- IQ can be used to define inquiry programs immediately.

1.1. Starting DATAMASTER

DATAMASTER consist of two parts which shows as two different icons in Windows:

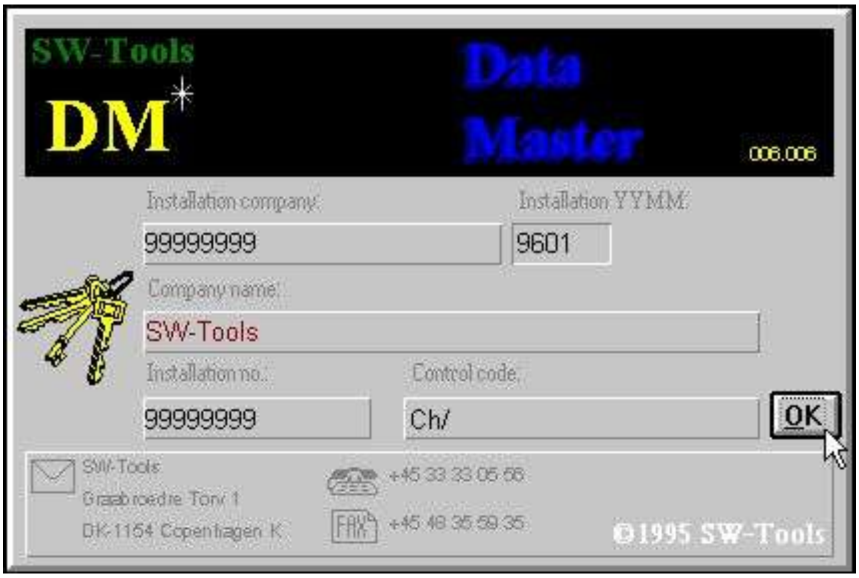


1. DATAMASTER icons

The file definition program is used to maintain the Data Dictionary, create new tables, amend/extend/delete existing tables and print database documentation.
The file maintenance program is used to define dataentry programs where you can create new records in the table, amend existing records or delete these.

1.1.1. Licence information

DATAMASTER is copyrighted SW-Tools and your licence information is shown briefly whenever the program is started.



Installation company: 99999999

Installation YYMM: 9601

Company name: SW-Tools

Installation no.: 99999999

Control code: Ch/

SW-Tools
Graabroedre Torv 1
DK-1154 Copenhagen K

☎ +45 33 33 05 56

FAX +45 48 35 59 35

©1995 SW-Tools

2. Licence screen

You are of course only allowed to use DATAMASTER according to your license agreement.

1.2. Principle of operation

The first part of DATAMASTER derives from the Data Dictionary module used for RAPGEN/IQ, this module has the facility to describe tables and has for DATAMASTER also the possibility to create the table itself. Apart from this every function described in the comprehensive Data Dictionary installation manual may be used but only a small subset of these is normally needed by the DATAMASTER user.

The second part of DATAMASTER is an extension of the IQ inquiry system enabling you also to make dataentry. This combination offers you the possibility to use all the search facilities in IQ together with your dataentry system.

1.2.1. Defining / redefining tables

A table is identified in the system by a ID consisting to letters or a letter and a digit. Whenever you want to use the table the ID must be given. The two table ID's SY and WW are reserved for special use, SY being SYSTEM FIELDS as today's date, WW being work fields for calculations in a program, each program will have is own work fields.

When you define a new table you just have to select a table ID, enter a name for the table and then for each field a name and a format, which determines the field length and layout. Each field is given a field number which are used to identify the field later on.

By ending the definitions DATAMASTER will install your new table definition in the data dictionary and create the table itself.

When you amend an existing table you can add new fields to the end of the table of change the name/format of existing fields. When finished DATAMASTER builds a new table and makes and runs a conversion program reading the records in the old and writing the new. When the conversion is successfully completed the old table is deleted and the new table activated.

You may insert/delete fields during this but note that fields are copied by field number as well as reports/programs may refer the table fields using the numbers. Each field may also be associated with check of valid values and an extensive documentation if so desired.

1.2.2. Making dataentry programs

As IQ, DATAMASTER enables you to make 3 different sort of programs:

- **One record maintenance**
- **List of records**
- **Transaction registration**

A program is defined just by entering the fields you want on the screen and may then be saved. The saved program contains the standard DATAMASTER screen layout and the calculations needed for reading/updating the tables involved.

You can freely amend such a saved program designing you own screen layout and tailoring the calculations to read/update any number of files you desire.

Calculations are parted into sections for e.g. making DEFAULT record by create, action upon input/click on a field and update of files so the standard program made by DATAMASTER is easy modified to meet your special requirements.

1.2.3. Database (ODBC) interface driver/ file type.

When defining a table you choose a file type among the installed database drivers Which driver you select depends on the future use of your tables.

Changing the file type during redefinition of a table causes DATAMASTER to CONVERT the file from one database system to another. The functionality of your programs is independent of which database systems are used unless the database itself has restrictions. The database driver must of course be able to insert/modify records in the tables.

The standard SSV (Semicolon Separated Variable length textfile) can be used for most DATAMASTER files, as this gives the smallest, fastest and most flexible file and may be imported into many other applications.

The ODBC drivers can be selected if you want to create files directly for a specific database system such as EXCEL, DBase, ACCESS, Informix, INGRESS or like. As these systems uses the far more complex SQL language you may experience that these are not as fast as the SSV files.

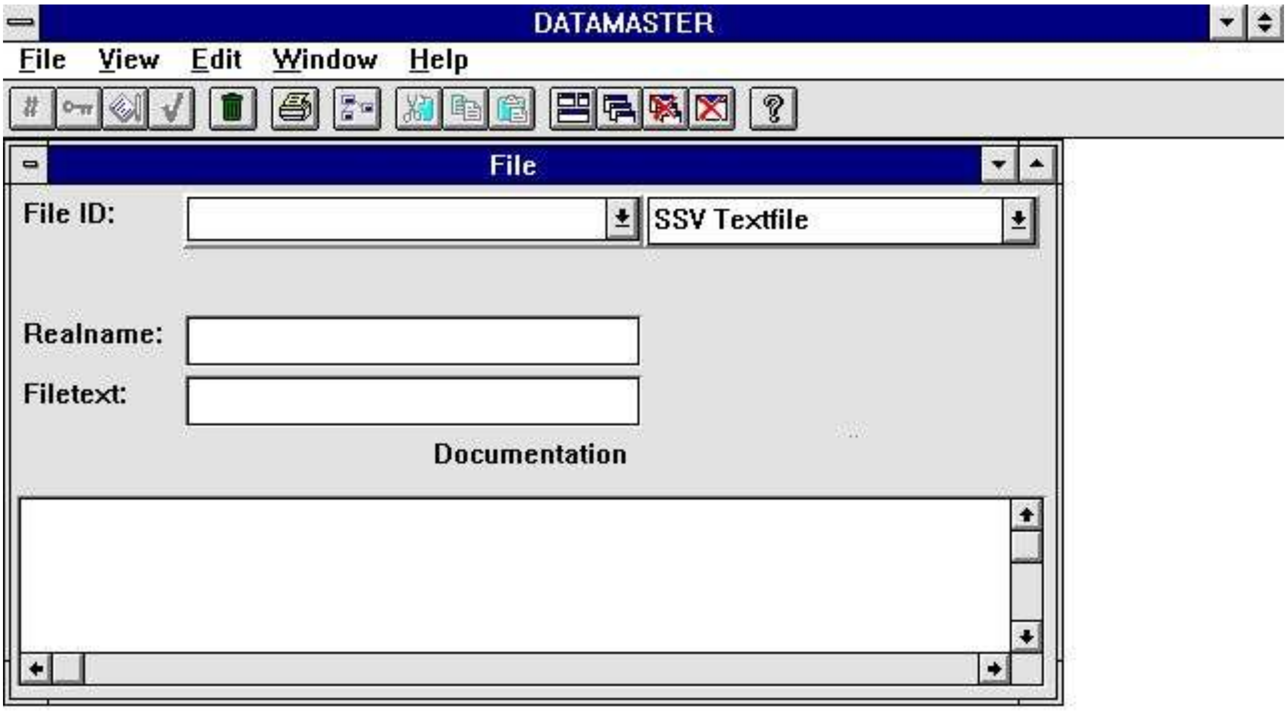
With the X-Basic driver using either the UNIX libraries or the CTRAS functions you can make basic indexed files directly.

2. Defining a new table

The Data Dictionary part of DATAMASTER enables you to define/redefine tables.
In this chapter we will create a simple customer table and just the functions necessary for doing this will be mentioned. Next chapter will describe how you puts data into this file.

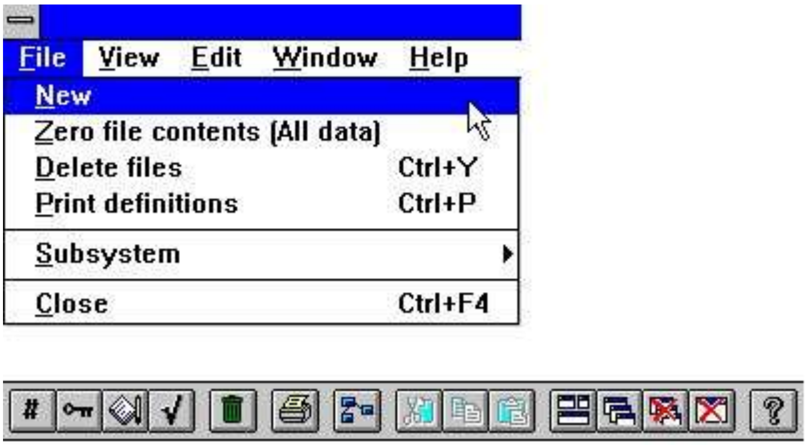
2.1. About the user interface

The Data-Dictionary always displays a main window with the following informations:



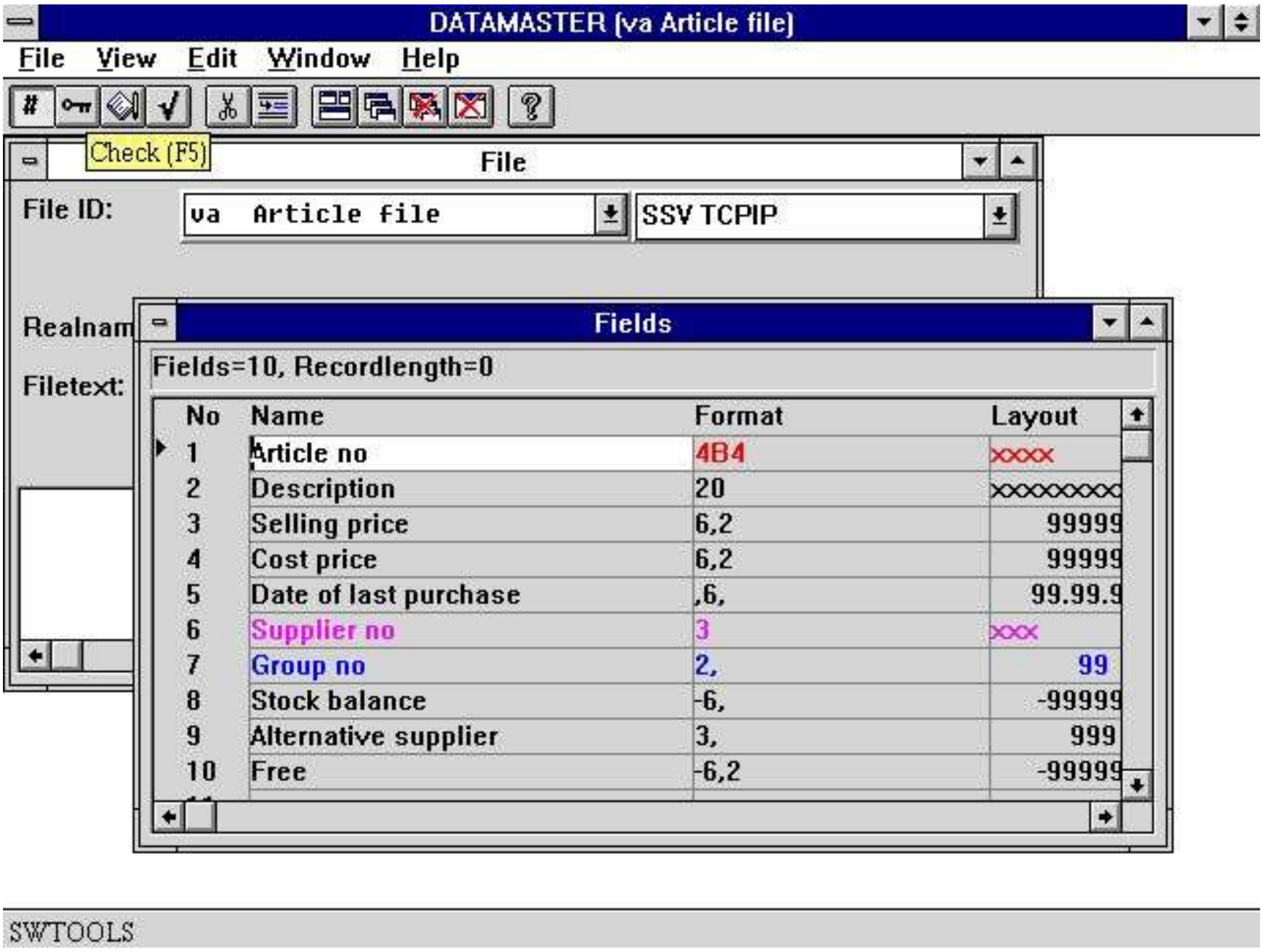
3. The main window

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



4. 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 area to function area, as for the pulldown menus. For example you will view this toolbar when activating the field function:



File ID: va Article file | SSV TCPIP

Realnam

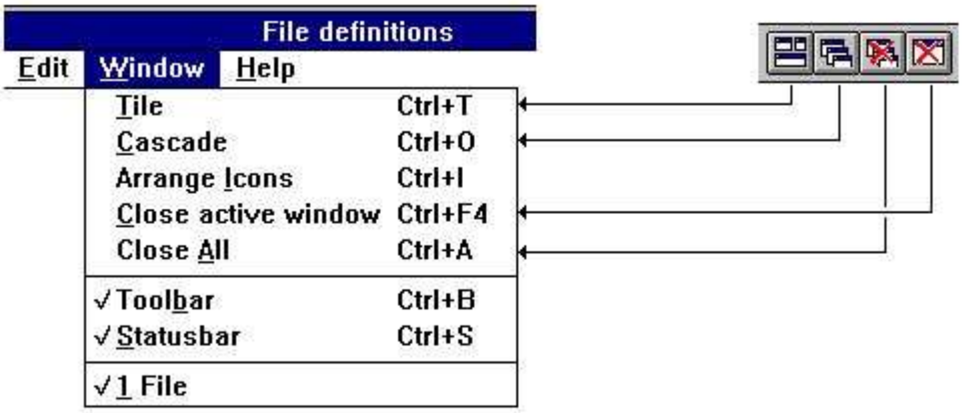
Filetext:

Fields=10, Recordlength=0

No	Name	Format	Layout
1	Article no	4B4	xxxx
2	Description	20	xxxxxxxxx
3	Selling price	6,2	99999
4	Cost price	6,2	99999
5	Date of last purchase	,6,	99.99.9
6	Supplier no	3	xxx
7	Group no	2,	99
8	Stock balance	-6,	-99999
9	Alternative supplier	3,	999
10	Free	-6,2	-99999

5. The toolbar during field input

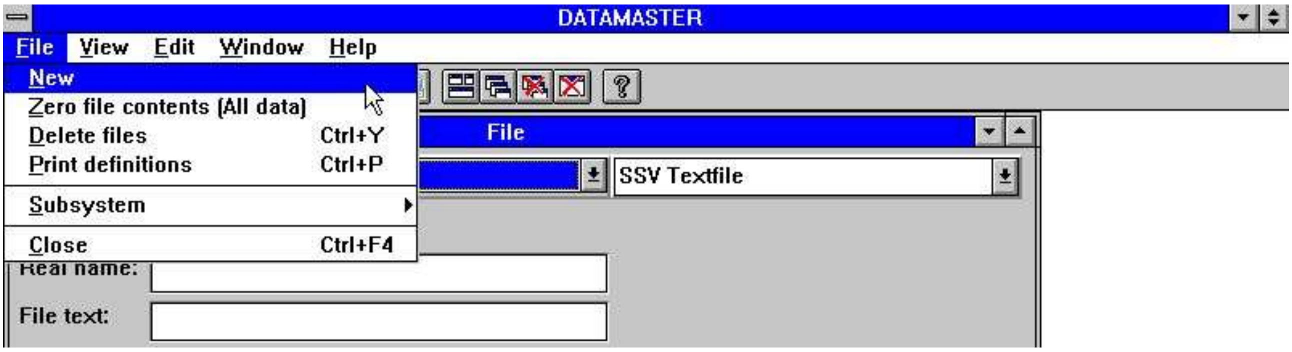
Even though the menus and toolbar buttons changes between functions the Data-Dictionary have some general functions attached on the toolbar. It is functions to place Windows, closing the active window and access to the on-line manual. Because these functions are attached to the toolbar they will of course also be accessible from the pulldown menus:



6. Generel menus and buttons

2.2. Creating a new table

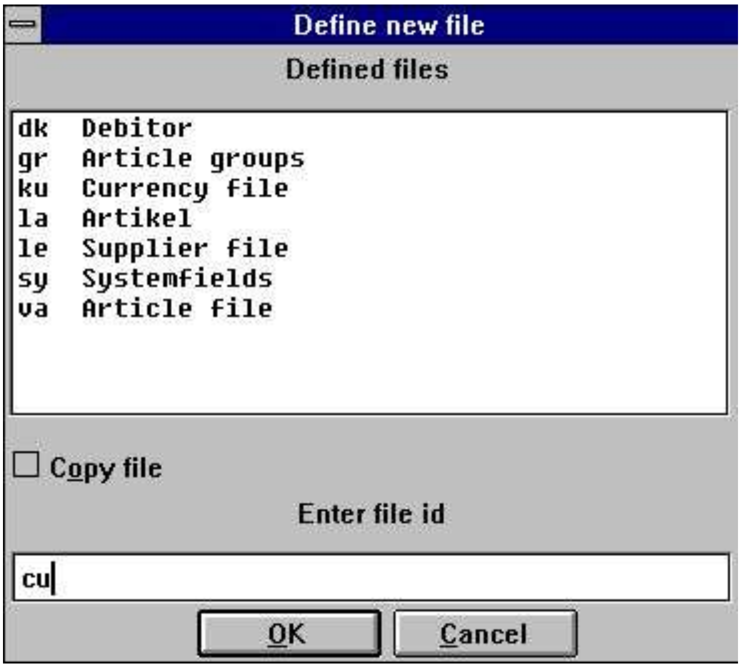
To create a new table you select the function NEW from the FILE menu.



7. Defining a new table

2.2.1. Table ID

You will be asked to enter a TABLE ID consisting of two letters or a letter followed by a digit:



dk	Debitor
gr	Article groups
ku	Currency file
la	Artikel
le	Supplier file
sy	Systemfields
va	Article file

Enter file id: cu

8. ID of new table

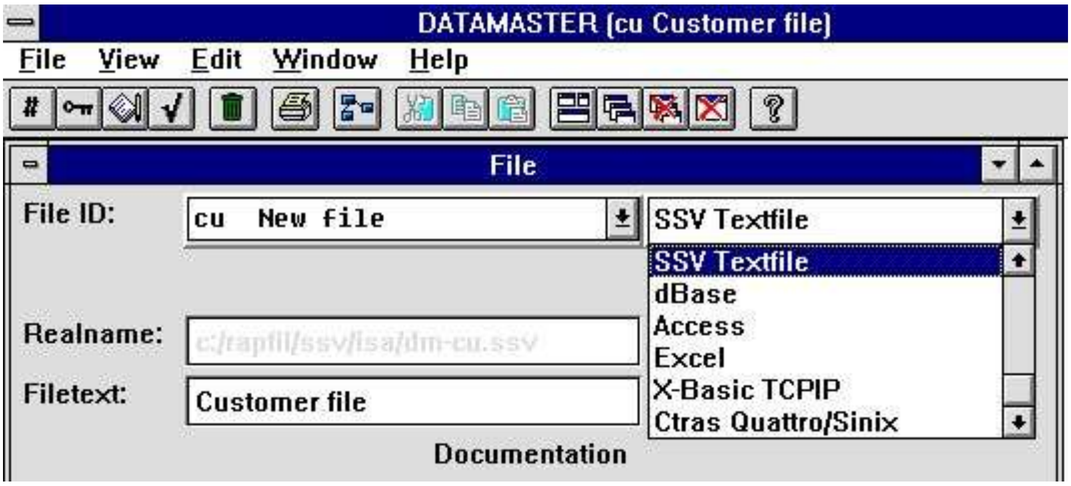
The ID's SY and WW are reserved for system use.
The listbox contains all tables used in the system - not only DATAMASTER tables. You must choose an ID which is not in use.

2.2.2. File name

After entering the ID the table name is defaulted to 'new file' and you should replace this with your desired name.
If you use TAB to move to next field the FIELD WINDOW will open allowing you to enter informations for each field.

2.2.3. Database interface (File type)

In case you have more database drivers installed you can select between these using the combobox:



File ID: cu New file

SSV Textfile

Realname: c:/rapfil/ssv/isa/dm-cu.ssv

Filetext: Customer file

9. Database interface (File type) selection

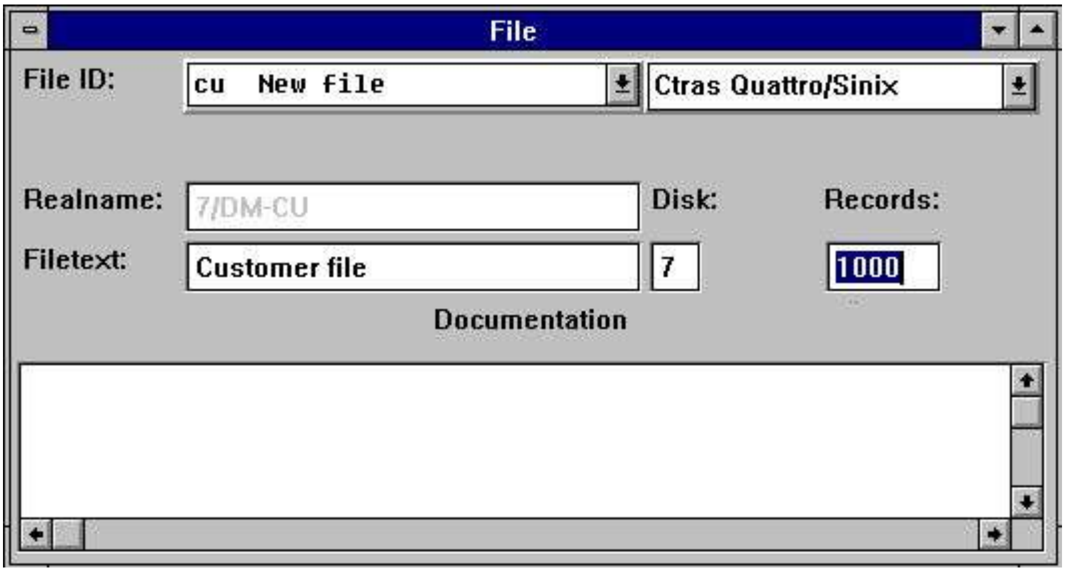
The chosen interface must be able to build and update tables in the database if not DATAMASTER will reject to create your table later on. Pls. refer to the discussion about which interface to choose in the introduction.

2.2.3.1. Real filename

Based on which database driver you have chosen a real filename will be generated by DATAMASTER and shown for your information. This may be a database table name, a DOS or UNIX filename or a BASIC filename.

2.2.3.2. Disk (lu)

If you have chosen a BASIC interface this field will be activated and you may place the file on any other disk instead of the default lu 0.



File ID: cu New file

Ctras Quattro/Sinix

Realname: 7/DM-CU

Disk: 7

Records: 1000

Filetext: Customer file

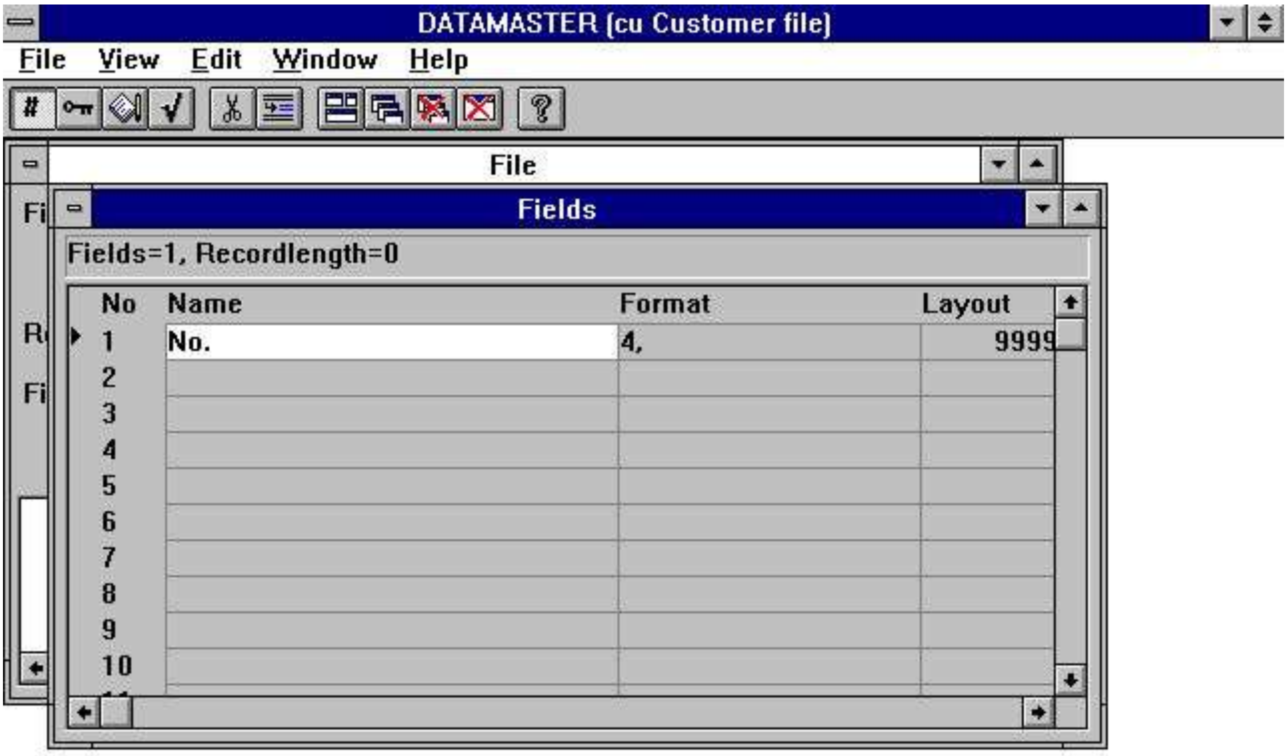
10. Disk (lu) and number of records

2.2.3.3. Number of records

Also for a BASIC interface you must specify the maximum number of records you want the file created with. The default is 1000.

2.3. Entering the fields

A table can consist of 1 to maximum 499 fields.



Fields=1, Recordlength=0

No	Name	Format	Layout
1	No.	4,	9999
2			
3			
4			
5			
6			
7			
8			
9			
10			

11. The field window

For each field you must enter a name and a format defining the length and type of the field. From the format specification DATAMASTER retrieves all information about the field needed for the selected database driver such as field pack type byte number if needed etc.

2.3.1. Field name

You can enter a free text of max. 256 characters and you may use special characters, language dependent letters and blanks as you desire.
We normally recommend to keep the field names short and clear as these are used as headings for screen programs and reports later on the ideal field name has the same length as the field itself.

2.3.1.1. Field SQL name

If the database driver you selected are using SQL DATAMASTER makes an unique and valid name from your free-text name. You do not have to worry about this.

2.3.2. Field format

The field 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 maximal length. Therefore the field can be defined as:

Format	**Description**
10	The field is limited to 10 characters
20	The field is limited to 20 characters

A numeric field contain only numeric values but may be defined with or without a decimal part. The following examples describes the syntax:

Format	**Description**
2,	The field may contain a value from 0 to 99
-2,	The field may contain a value from -99 to 99
5,	The field may contain a 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:

Format	**Description**
,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

2.3.2.1. Editing of numeric fields

A numeric field value can also be edited. If the values has to be printed or displayed with a comma or dot for each 1000 of the integer part just insert a leading comma in the format

,9,2

The value 123456789.12 will be displayed as

123,456,789.12

Please note that date fields with the format ,6, and ,8, cannot be used for other values than dates.

A field defined with leading comma will always support negative values as if defined with leading sign, e.g.

-,9,2 equals ,9,2

The character used to edit the field, in this example a comma, may be changed with the Preferences function.

Leading zeros or stars is also an option for numeric fields. If the field format contains the character &, e.g.

9,2&

the value 12345.12 is displayed as

000012345.12

or if the format is defined with the character *, e.g.

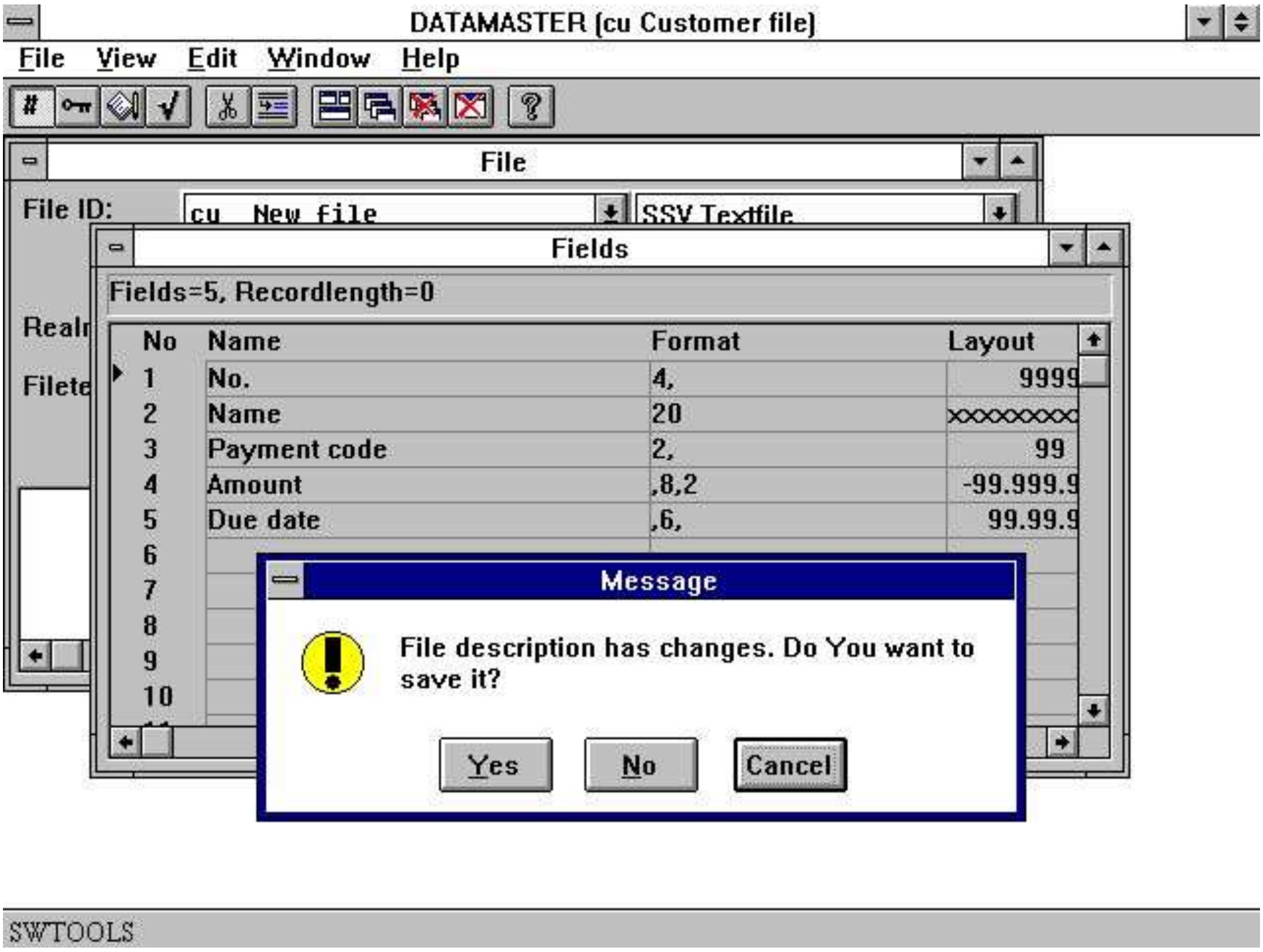
9,2*

the value 12345.12 is displayed as

*****12345.12**

2.4. Ending the definition

When you close the file definition [ALT+F4] or choose another file you will be asked if your definition should be kept.



Fields=5, Recordlength=0

No	Name	Format	Layout
1	No.	4,	9999
2	Name	20	xxxxxxxxx
3	Payment code	2,	99
4	Amount	,8,2	-99.999.9
5	Due date	,6,	99.99.9
6			
7			
8			
9			
10			

Message

File description has changes. Do You want to save it?

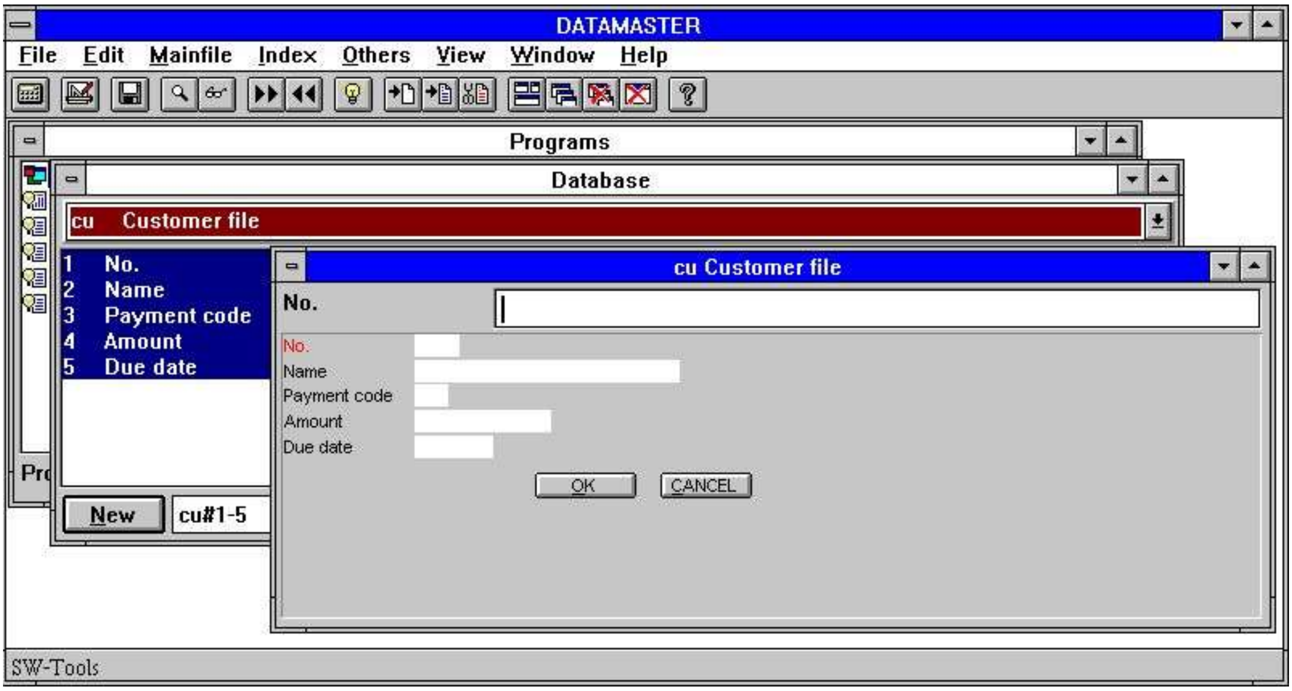
Yes   No   Cancel

12. Ending the definition

When you answer YES the definition is stored and the file itself will be build and initialised according to the database driver you have selected.

3. Entering data into the table

Defining a dataentry program with DATAMASTER is just like defining a query in IQ but you will be able to make input to the fields.



13. The first dataentry program

The newly created table (cu) is present in the Data Dictionary and you can just define a query program consisting of all fields cu#1-5. As you have selected DATAMASTER this becomes a dataentry program.

The first field (no.) is marked in red. DATAMASTER has made this field to the main index keyfield as you have not told anything else.

3.1. DATAMASTER and IQ

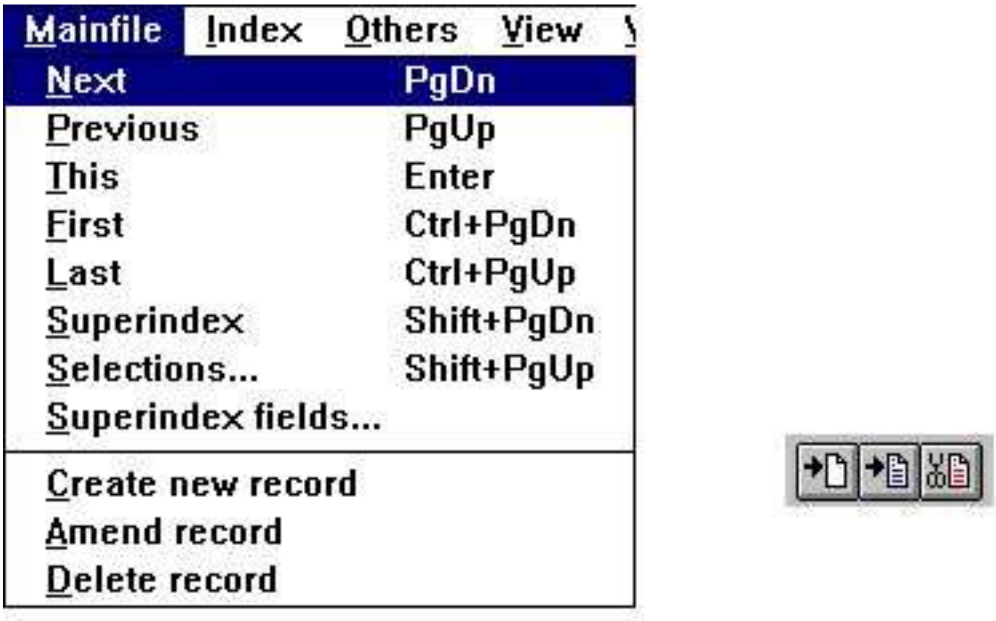
If you have bought and have a licence code for IQ also you may use all the functions described in the IQ manual on your new file.
Without a IQ licence you can only make programs involving you own DATAMASTER files. You will be able to enter data into your file(s), make programs with calculations and reading of other DATAMASTER files but you cannot use the IQ extensive search facilities:

- **First / Last / Previous record**
- **First / Last / Previous / Reverse order of transactions**
- **Superindex**
- **Selections**
- **Index selection lock**
- **Display key**
- **Case dependent search**
- **Inter-program communication**

In the rest of this manual we will assume that you have IQ active so no restrictions in use occurs.

3.2. The DATAMASTER functions

A DATAMASTER program is extended with the functions for create/amend/delete a record in the main file, each function being associated with an icon on the toolbar.

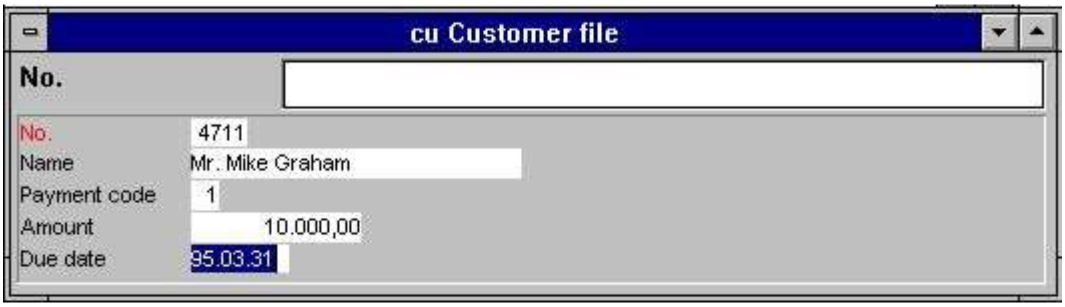


14. DATAMASTER mainfile menu and the function icons

3.2.1. Create a new record

When you select 'create new record' the input field changes from the search key to the first key field defined in your file (no.).
You must enter the key where after DATAMASTER makes a check that this key is not present in the file already. Hereafter you enter the data for each field.
You may use

- TAB or just ENTER to move to next field,
- SHIFT+TAB to go to previous field,
- CURSOR UP/DOWN to move straight up/down in the fields
- click on any field with the mouse to jump directly to this



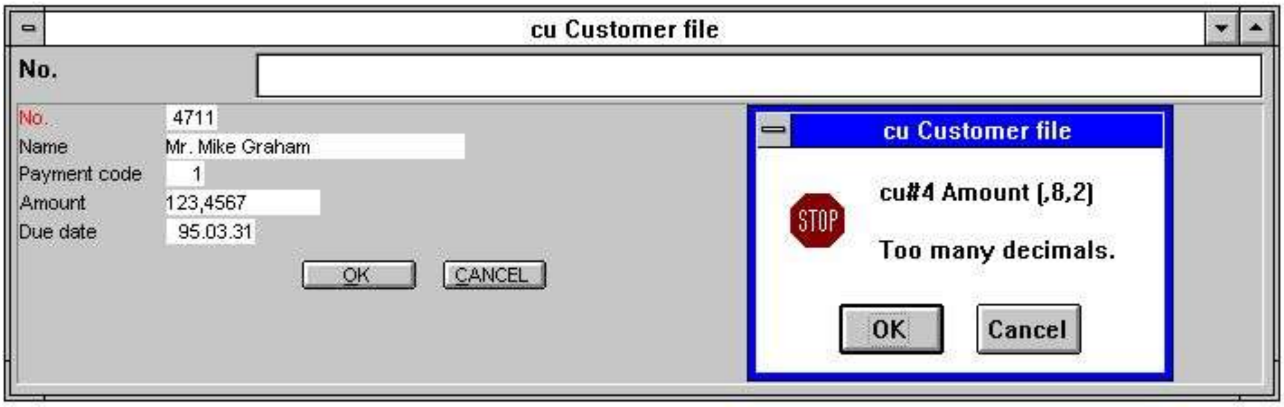
No.

No.	4711
Name	Mr. Mike Graham
Payment code	1
Amount	10.000,00
Due date	95.03.31

15. Dataentry

3.2.1.1. Input validation and date fields

During your input each field is validated so it will not exceed the field length you have specified, numeric fields must contain digits and date fields must contain a valid date or zero. If not you will get an error message box like:



cu Customer file

No.

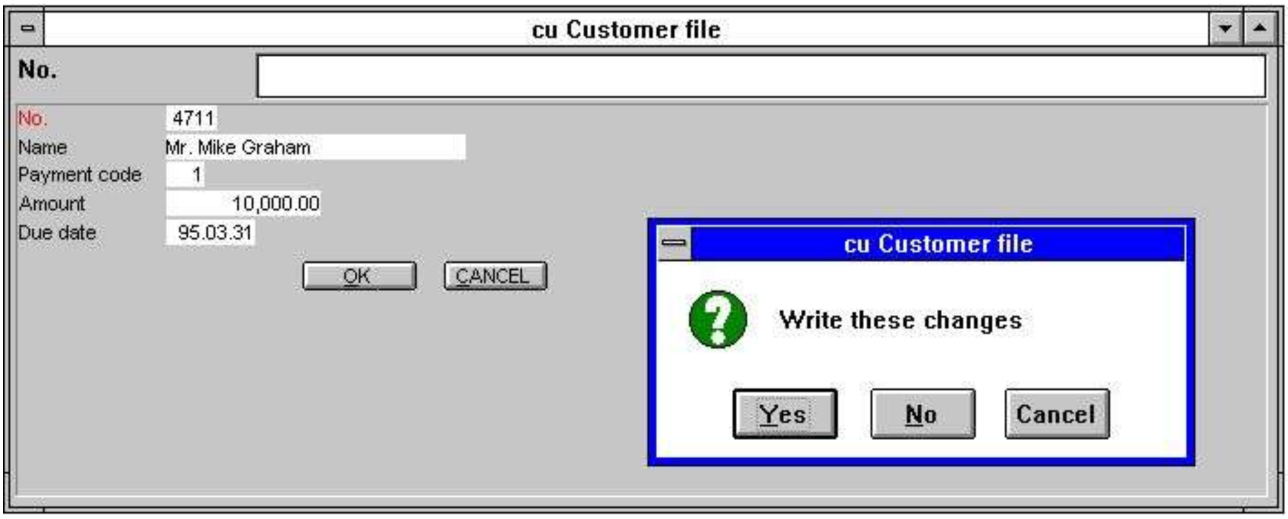
No. 4711
Name Mr. Mike Graham
Payment code 1
Amount 123,4567
Due date 95.03.31

cu#4 Amount [,8,2]

Too many decimals.

16. Input validation error message box

Date fields may be entered as DDMMYY, YYMMDD or YYYYMMDD and will automatically be turned according to the date specification by installation. You may also enter just a part of the date as DD or DDMM - the missing part will be taken from the current system date.
When you have finished the input for this record you may select the 'Create new record' function once more or just press ESC. You will be asked to confirm writing the record to you file:



cu Customer file

No.

No. 4711
Name Mr. Mike Graham
Payment code 1
Amount 10,000.00
Due date 95.03.31

Write these changes

17. Confirmation of file update

3.2.2. Amending a record

To amend a record you can first look this up by entering a value in the search key and using the IQ search functions and then choose the AMEND icon or just press TAB to jump from the keyfield to the first non-key field in your file.

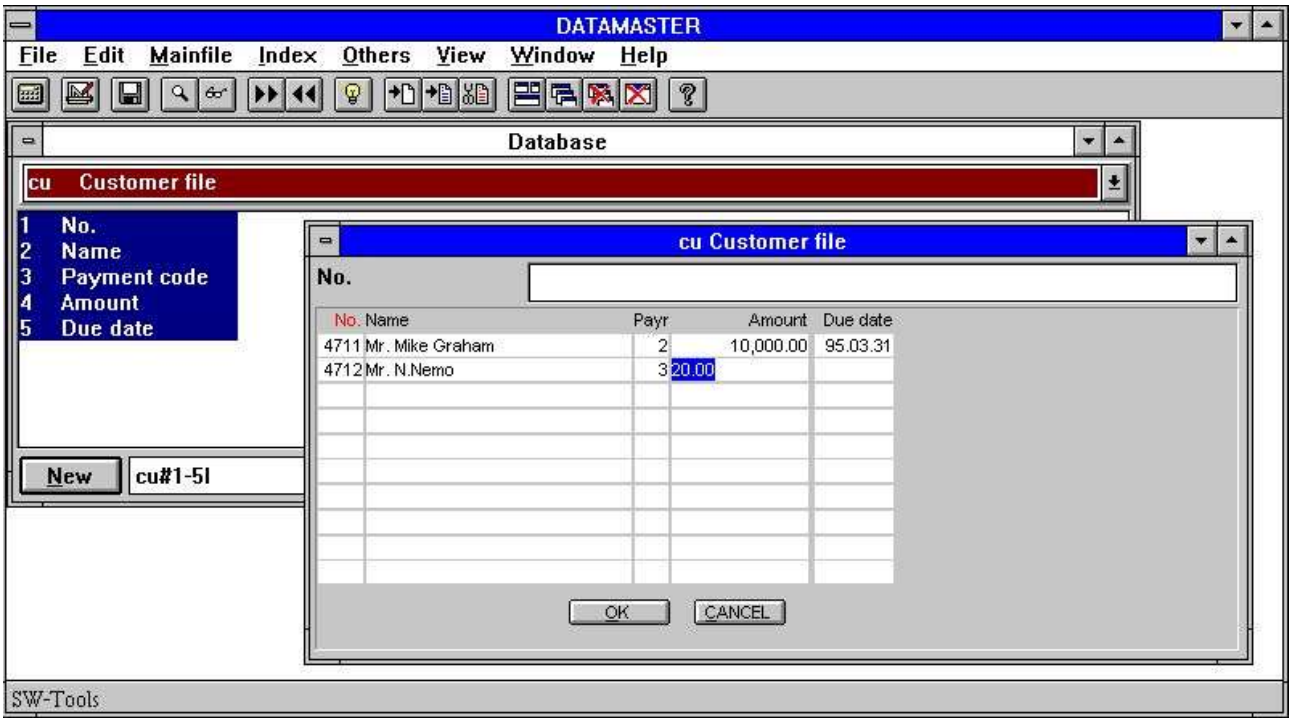
You may change all fields including the key fields. You finish the amend function by use of ESC to jump to the keyfield or any of the functions to select another record as PAGE DOWN for next record. Also here you will be asked to confirm writing the changes for each record.

3.2.3. Deleting a record

When you have looked up a record just choose the DELETE icon. You will be asked to confirm the removal of the record from the file.

3.3. Dataentry on list form

As in IQ you can make a list of records by placing the letter l after the fields.



cu Customer file

1 No.
2 Name
3 Payment code
4 Amount
5 Due date

cu#1-5l

cu Customer file

No.

No.	Name	Payr	Amount	Due date
4711	Mr. Mike Graham	2	10,000.00	95.03.31
4712	Mr. N.Nemo	3	20.00	

18. Dataentry on list form.

In this way you will be able to search/select a bunch of records, amend these or insert/delete records among these.

The records in such a list is kept in an internal buffer and all your amendments goes to this until you finishes the page by PAGE DOWN for next page or by ESC to jump to the keyfield. At that moment you will be asked to confirm all the updates.

3.3.1. Create a new record

Just click on a blank line and type your data. You may also choose the 'create new record' icon when positioned on a line. This causes a new empty line to be inserted ready for new input.

3.3.2. Amending a record

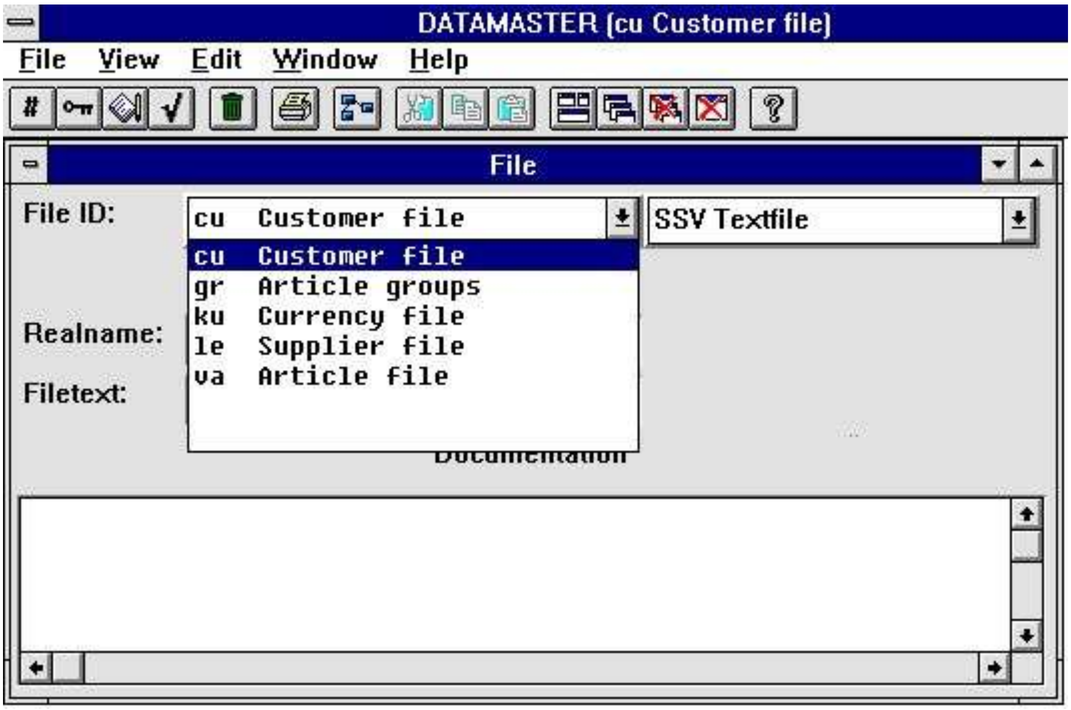
Just click on a line containing data and type your the new values.

3.3.3. Deleting a record

When you are positioned on a line containing data click on the 'delete' icon and the line are removed completely from the screen.

4. Altering and extending a table definition

When you modify a DATAMASTER table you can change all informations in the definition except the to letter FILE ID identifying the file.
After the modifications when you confirm your changes the new definition will be stored not overwriting the present definition but in a file prefixed NEW. The old and the new file definition will both be present at this time.
DATAMASTER now compares the old and the new definition to see if your modifications requires changes in the file itself. If so a NEW file is build and a conversion program generated and run copying all data from the old file to the new file.
If everything goes ok the old definition and the old file will be deleted and the new file renamed (activated). If something goes wrong the old definition and the old file will be kept as if no changes had occurred. Also your NEW definition with the modifications will be kept - next time you try to modify the definition possible after freeing disk space for the new file on the server you will be asked whether the new definitions should be activated.
To amend the definition for a file open the box with FILE ID and select the table you want to modify.



File ID: cu Customer file

SSV Textfile

cu Customer file
gr Article groups
ku Currency file
le Supplier file
va Article file

Realname:

Filetext:

19. Selecting the table to amend

Note the listbox contains DATAMASTER files only - you cannot alter other tables in your system due to security reasons.

4.1. Changing the database interface (file type)

If you change the file type the table will be converted from one database system to another. In this way a file created e.g. with the fast SSV interface can be converted into any other database system as ACCESS.

4.2. Modifying the field informations.

You may freely modify the field names and the field formats.

4.2.1. Field name

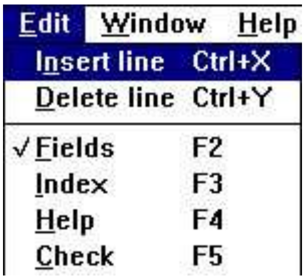
If the database driver uses SQL names a conversion from old SQL name to new will be done when you finishes your modifications.

4.2.2. Field format

A field can be extended to hold more characters or digits without problems. If you shorten a field to hold fewer characters it will be your own responsibility that the present data can fit into the new field, if not the values will be truncated.
Numeric fields may be changed to textfields and textfields may be changed to numeric also. DATAMASTER will do its best to convert your values.

4.2.3. Inserting/deleting fields

In the EDIT menu and on the toolbar you will find the commands/icons to insert/delete fields.



20. The EDIT menu

When you insert or delete fields the index definitions, field help documentation and check codes (see later) automatically moves with the new field numbers.
However you should be careful with these commands as the conversion program DATAMASTER generates uses the field numbers, it is

- NEW field 1 = OLD field 1,
- NEW field 2 = OLD field 2,
- NEW field 3 = OLD field 3...

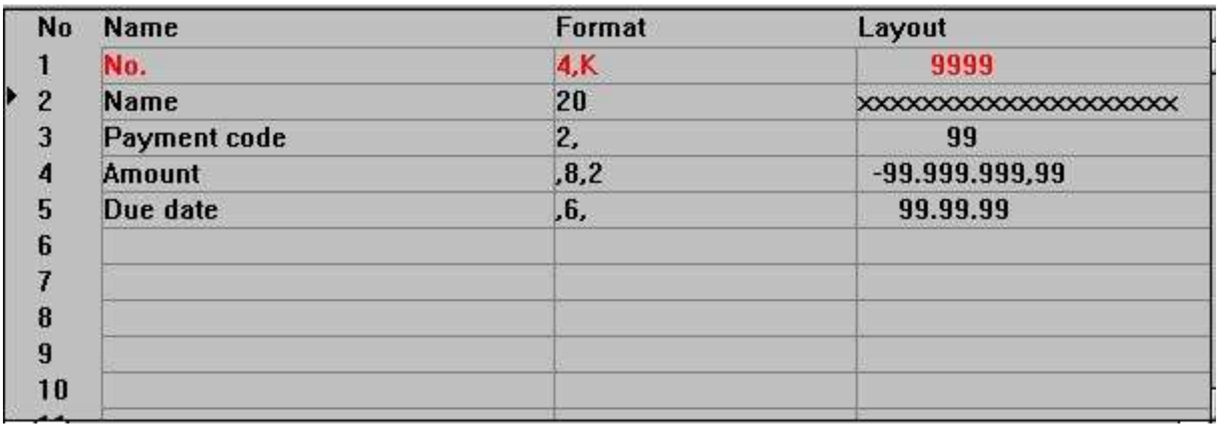
Also your defined programs for dataentry, IQ or RAPGEN may refer to field numbers, these programs you must change manually afterwards.
Normally insert/delete fields should be used only for empty files with no or very few defined programs.

4.2.3.1. Free field numbers

When you define a program using DATAMASTER, IQ or RAPGEN a number of free fields, default 40, are added to the mainfile definition. If the main file has 10 fields you refer these free fields as field number 11-50.
If you extend the file to say 20 fields the free field numbers moves to 21-60.
In the calculations/layouts of the programs you have already defined the free field numbers will move automatically as these are stored not as e.g. #11 but as WW#1.

4.2.4. Marking fields as index fields

Index defines how the data is sorted in a table and how to access a specific record. You use the index e.g. when searching a record in IQ.
When you create a new table the first field is automatically named to be the main (primary) unique key (index) for the table unless you tell something else.
You see this when amending the file as the first field is shown in red and a K is added to the format (unique Key).



No	Name	Format	Layout
1	No.	4,K	9999
2	Name	20	xxxxxxxxxxxxxxxxxxxx
3	Payment code	2,	99
4	Amount	,8,2	-99.999.999,99
5	Due date	,6,	99.99.99
6			
7			
8			
9			
10			

21. The field K index mark

4.2.4.1. K mark for unique Key fields

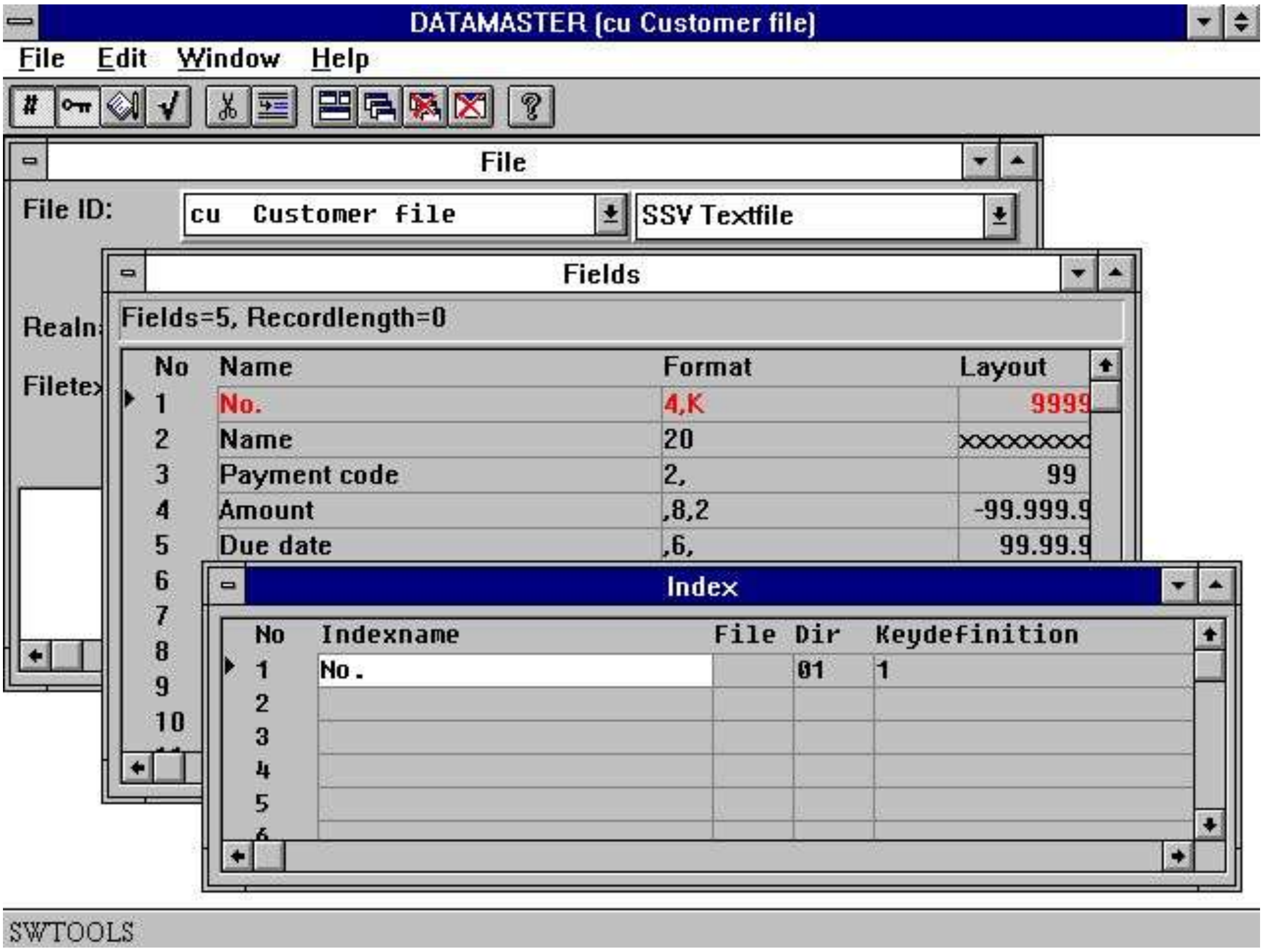
You can mark other fields with K making these unique key fields and you can remove the K mark on the first field.
The K mark is normally used only for the first field as e.g. customer number identifying the record. Two customers can not have the same number.
When the file contains data you should be careful marking fields with K as the data in the file cannot be copied unless they really holds different values for each field.

4.2.4.2. D mark for Duplicates allowed

You may also mark fields with D to allow more records with the same field value. The D mark is normally used when you wants to create a matchcode or a name index.

4.3. Index definitions

You may define other index already when creating the table or you may add / delete indexes later on. The conversion program generated will take hand of these modifications also.
A maximum of 49 index of each 128 characters length each can be defined for a table if the database driver allows this. (Basic allows only 15 index of 30 characters each).
To edit indexes for a table select the following function:



File ID: cu Customer file | SSV Textfile

Fields=5, Recordlength=0

No	Name	Format	Layout
1	No.	4,K	9999
2	Name	20	xxxxxxxxx
3	Payment code	2,	99
4	Amount	,8,2	-99.999.9
5	Due date	,6,	99.99.9
6			
7			
8			
9			
10			

No	Indexname	File	Dir	Keydefinition
1	No.		01	1
2				
3				
4				
5				

22. Editing index information

One index definition equals one line on the screen with the following informations:

4.3.1. Name

The index name may contain all letters, digits, blanks and special characters. It is used as description when e.g. IQ displays a listing of possible index selections for search.
If the database driver uses SQL names this free text name is converted into an unique SQL name also with no special characters.

4.3.2. Fileid

The fileid is only used when defining relations from this table to others, see later. Always leave this field blank when defining an index.

4.3.3. Directory (Index number)

The directory, also known as the index number, is a two-digit value. The index definitions must be numbered from 1 an upwards, you are not allowed to have to lines with the same number.

4.3.4. Keydefinition

The keydefinition is the real definition of the index telling which fields should be part of this. A few examples:

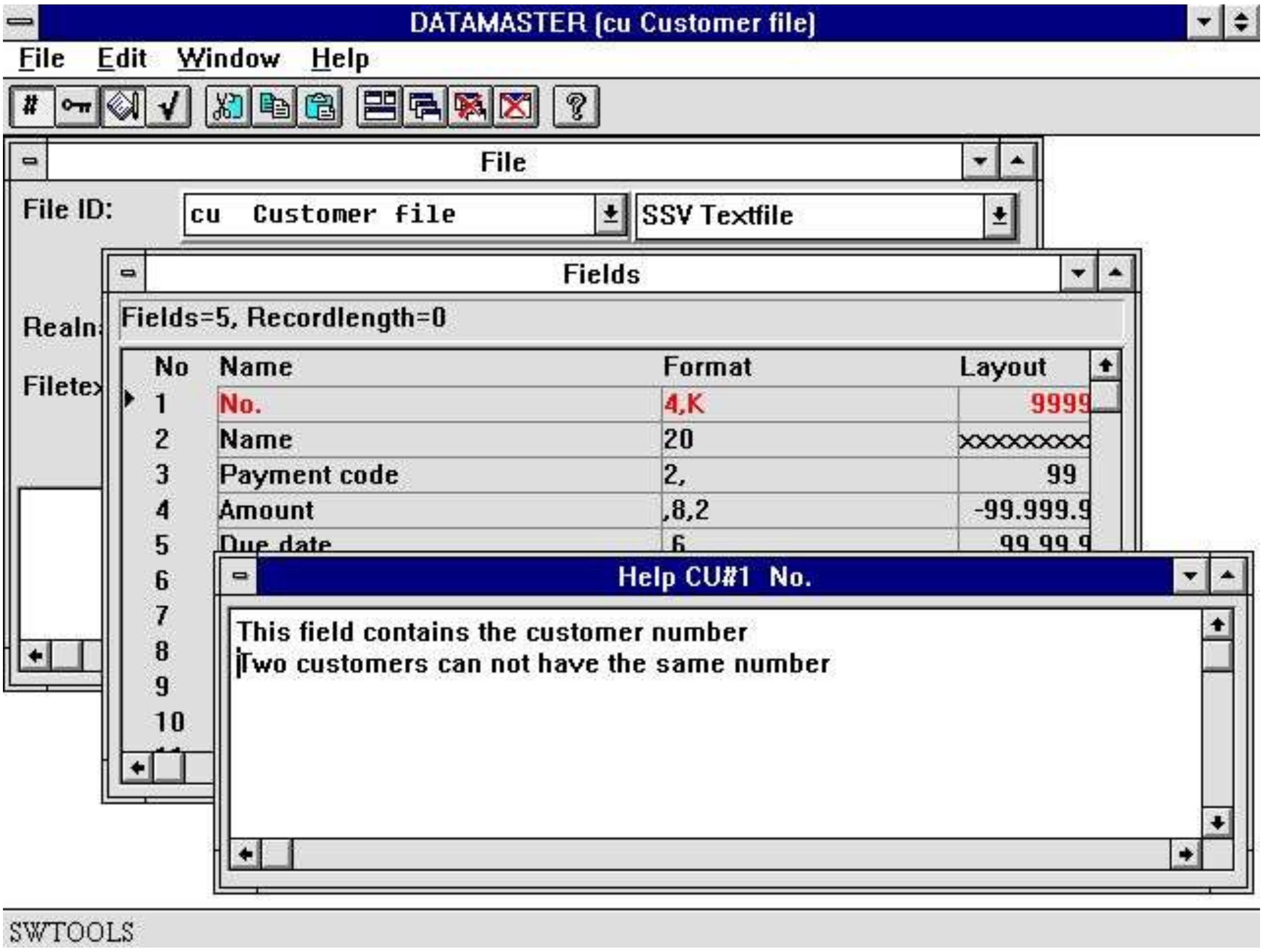
Keydefinition	**Description**
1	Field 1 is an unique key
#2	Field 2 is an unique key, # can be given or omitted as desired
2,1	Field 2 is an key made unique by adding the main keyfield 1
2(5,10)	Field 2 character 5 to 10 (6 characters) is a unique key
2,NP	Field 2 is a key which may have duplicate values, NP indicates duplicates allowed.

4.4. Table and field documentation (Helptext)

For each of your tables you can enter a documentation text by click on the documentation field on the file page.

4.4.1. Entering the fielddocumentation

Also for each field you may enter a free text of maximum 99 lines when you open the field documentation window from the EDIT menu or by use of the icon.



File ID: cu Customer file | SSV Textfile

Fields=5, Recordlength=0

No	Name	Format	Layout
1	No.	4,K	9999
2	Name	20	xxxxxxxx
3	Payment code	2,	99
4	Amount	,8,2	-99.999.9
5	Due date	6	99 99 9
6			
7			
8			
9			
10			

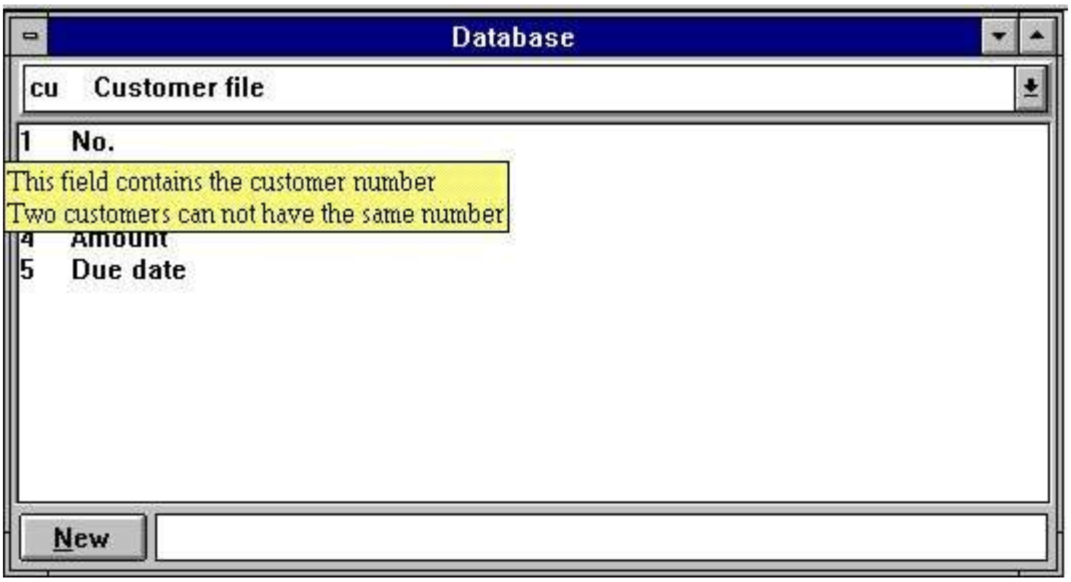
Help CU#1 No.

This field contains the customer number
Two customers can not have the same number

23. Editing field documentation

4.4.2. Floating on-line field help

The documentation will be printed when you print your file definitions and whenever you define a program in DATAMASTER, IQ or RAPGEN the database window for selecting fields will display floating help when the cursor is moved over the field.



This field contains the customer number
Two customers can not have the same number

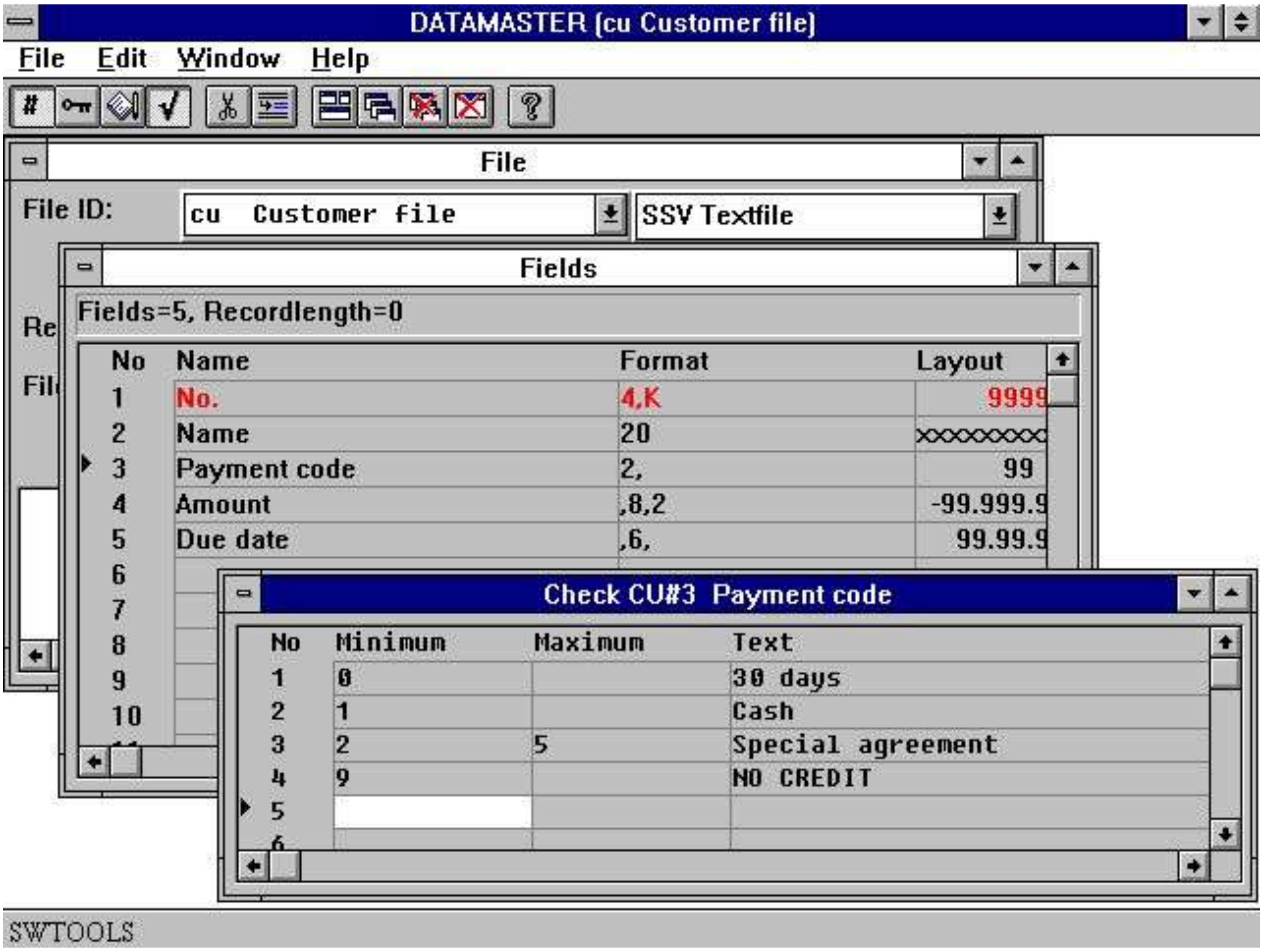
24. Floating on-line field help

4.5. Field checkcodes

Field checkcodes are used to define allowed values for fields. You can associate each field with 499 different ranges of values each connected to a description text. The checkcodes are designed for 'reasonable' small number of codes - if you need more you should create a DATAMASTER table holding the codes instead.

4.5.1. Defining the check

You open the check window from the EDIT menu or by click on the toolbar icon where after you enters the range value from-to and the text associated with that range.



Fields=5, Recordlength=0

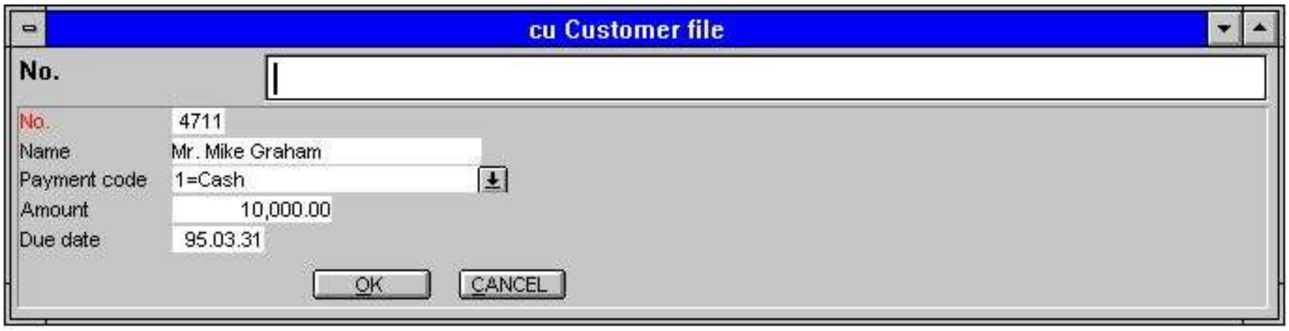
No	Name	Format	Layout
1	No.	4,K	9999
2	Name	20	xxxxxxxx
3	Payment code	2,	99
4	Amount	,8,2	-99.999.9
5	Due date	,6,	99.99.9
6			
7			
8			
9			
10			

No	Minimum	Maximum	Text
1	0		30 days
2	1		Cash
3	2	5	Special agreement
4	9		NO CREDIT
5			

25. Defining check for a field

4.5.2. Field validation

When you define a DATAMASTER dataentry or IQ query program (as cu#1-5) an additional field is place behind all fields with associated check defined with room for the text. The length of this textfield is taken as the maximum length of any text you have defined.



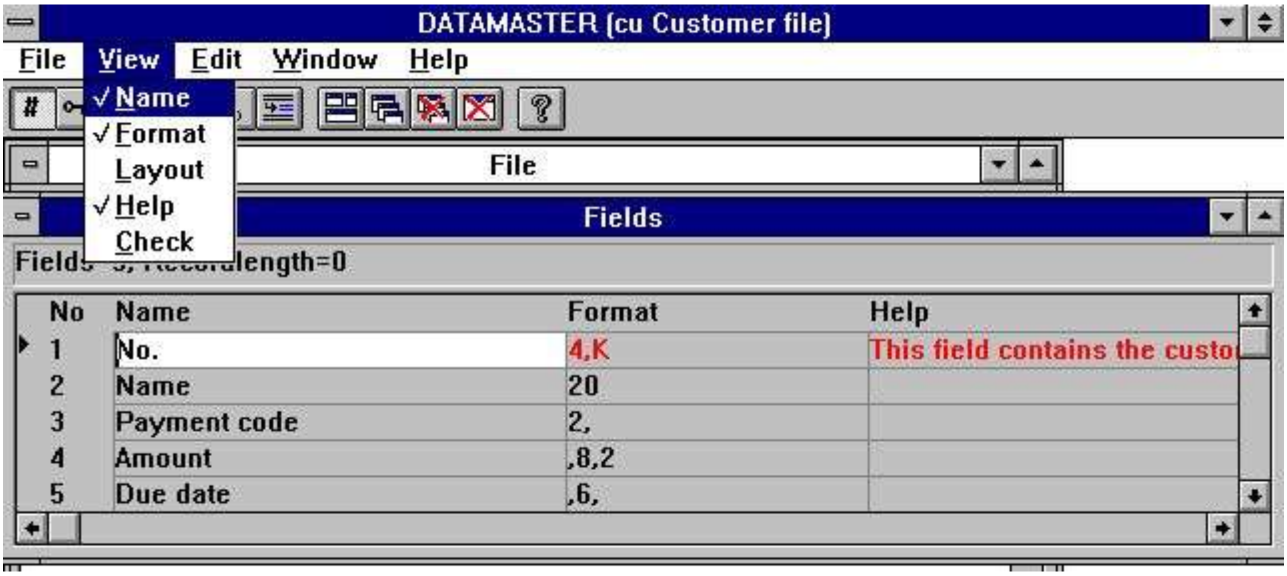
No.	4711
Name	Mr. Mike Graham
Payment code	1=Cash
Amount	10,000.00
Due date	95.03.31

26. Field validation

The field will be validated so only the ranges you defined can be entered.
If you have created data in the file before defining the check the associated text will also be shown for these records but there may occur values which are not longer valid. These are displayed without text.

4.6. The view menu

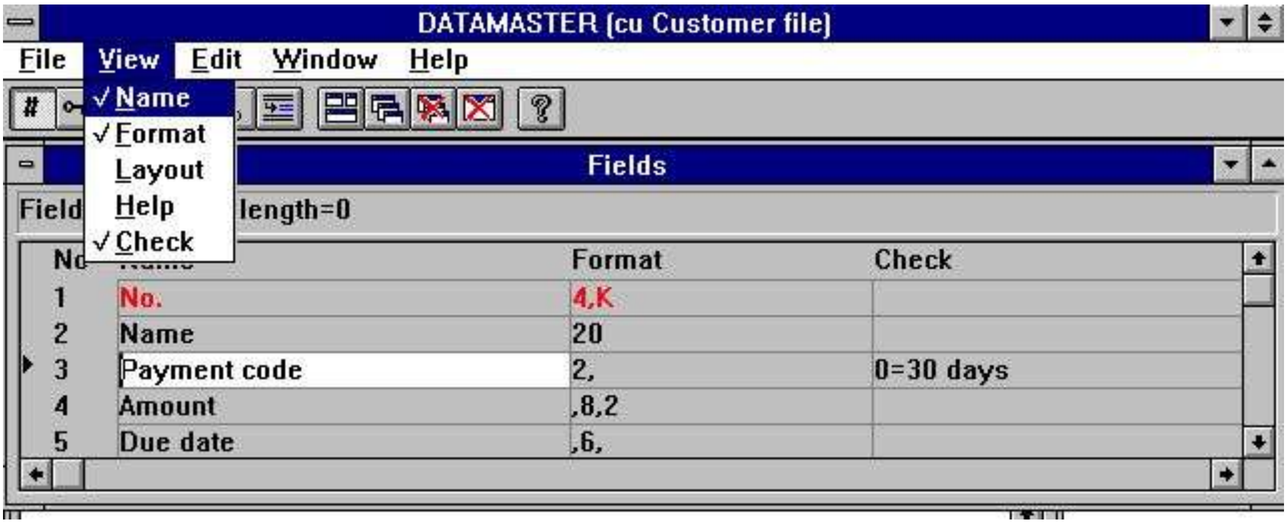
Using the VIEW menu you can replace the layout column for each field with either the first documentation line or check code line. This gives you a fast overview of which fields have got/have not got any definitions.



Fields ... length=0

No	Name	Format	Help
1	No.	4,K	This field contains the custo
2	Name	20	
3	Payment code	2,	
4	Amount	,8,2	
5	Due date	,6,	

27. Overview of field documentation



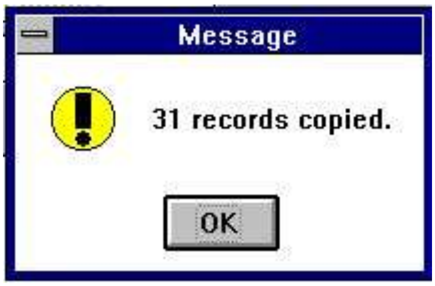
Field ... length=0

No	Name	Format	Check
1	No.	4,K	
2	Name	20	
3	Payment code	2,	0=30 days
4	Amount	,8,2	
5	Due date	,6,	

28. Overview of field check

4.7. Ending the redefinition

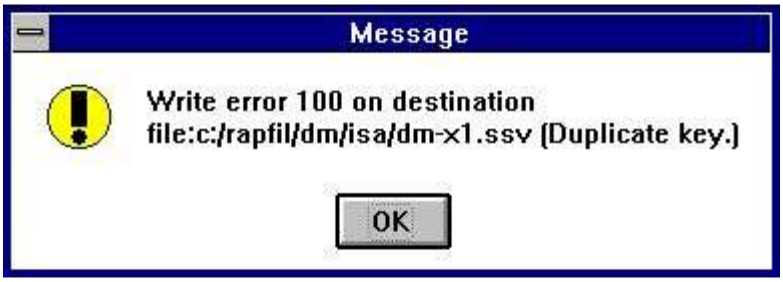
When you finishes your modifications of the definition the table will be converted to the new record layout if needed and you will get a message like:


Message

31 records copied.

29. Confirmation of table conversion

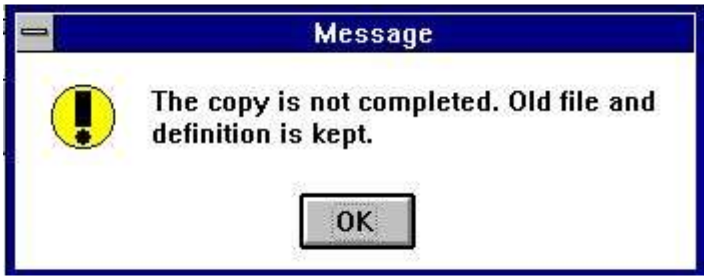
If something goes wrong like the server runs out of disk space or you have defined an unique index where the same value of the keyfield occurs you will get a message like:


Message

Write error 100 on destination file:c:/rapfil/dm/isa/dm-x1.ssv (Duplicate key.)

30. Error message from the conversion program

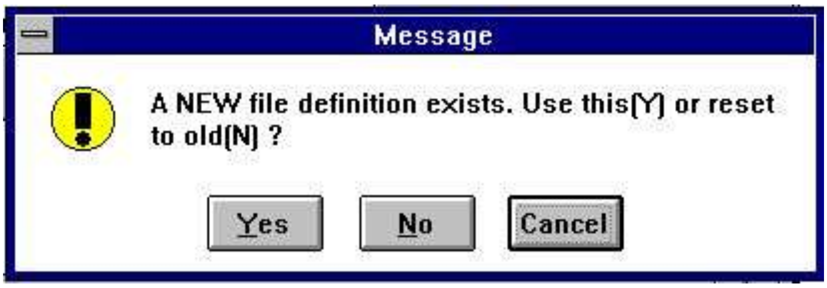
followed by:


Message

The copy is not completed. Old file and definition is kept.

31. End message from the conversion

Next time you try to modify the table you have the possibility to activate or ignore your modifications and try again:


Message

A NEW file definition exists. Use this(Y) or reset to old(N) ?

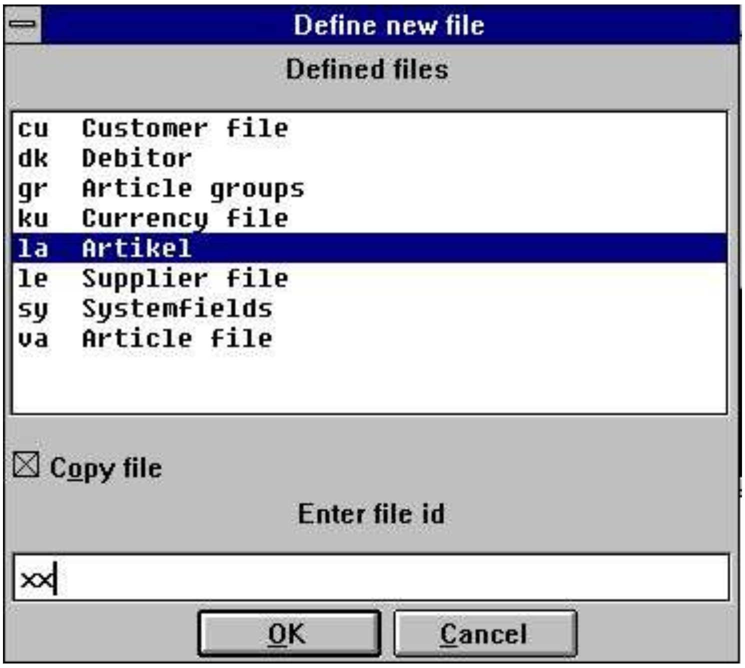
32. Try again

If you answer Y your modified definition will be used, N causes the modifications to be ignored. The following table conversion will always be done based on the original definition and table.

5. Copying, deleting and printing the definitions

5.1. Copying a table

To copy a table select NEW from the file menu and in the listbox shown:



Define new file

Defined files

```
cu  Customer file
dk  Debitor
gr  Article groups
ku  Currency file
la  Artikel
le  Supplier file
sy  Systemfields
va  Article file
```

☒ Copy file

Enter file id

xx

33. Marking a table for copy

select the source table by click on this, mark the copy box and enter a new ID.
The complete table definition will be copied and you can modify this. When you end your modifications all data in the original table will be copied to the new DATAMASTER table.

5.1.1. Copying a definition only

To copy a definition but not data do exactly as above but check also the option 'Zero file contents' in the file menu (see below).

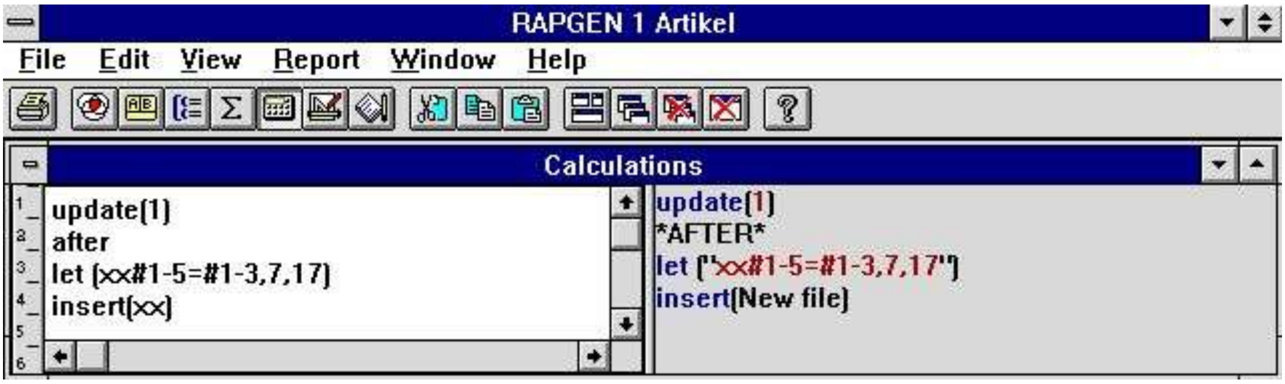
5.1.2. Data conversion between systems

As the listbox above does not only contain DATAMASTER files you can select any file in the system as source file - in the above example the COMET article file from X-Basic is selected as input.

If you for the new DATAMASTER table selects another file type, e.g. ACCESS, and then just ends the definition, all data in the input file is converted into an ACCESS database table.

5.1.3. Using RAPGEN to copy data

The file you define in DATAMASTER may have a quite different set of fields than in the original file. In this case you should not choose the copy function but just define a new - empty - DATAMASTER file and then a report with the source file (la) as main file inserting records in the DATAMASTER file using calculations like:



```
update(1)
after
let (xx#1-5=#1-3,7,17)
insert(xx)
```

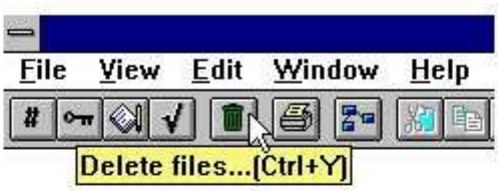
```
update(1)
*AFTER*
let ('xx#1-5=#1-3,7,17')
insert(New file)
```

34. RAPGEN calculations for data conversion

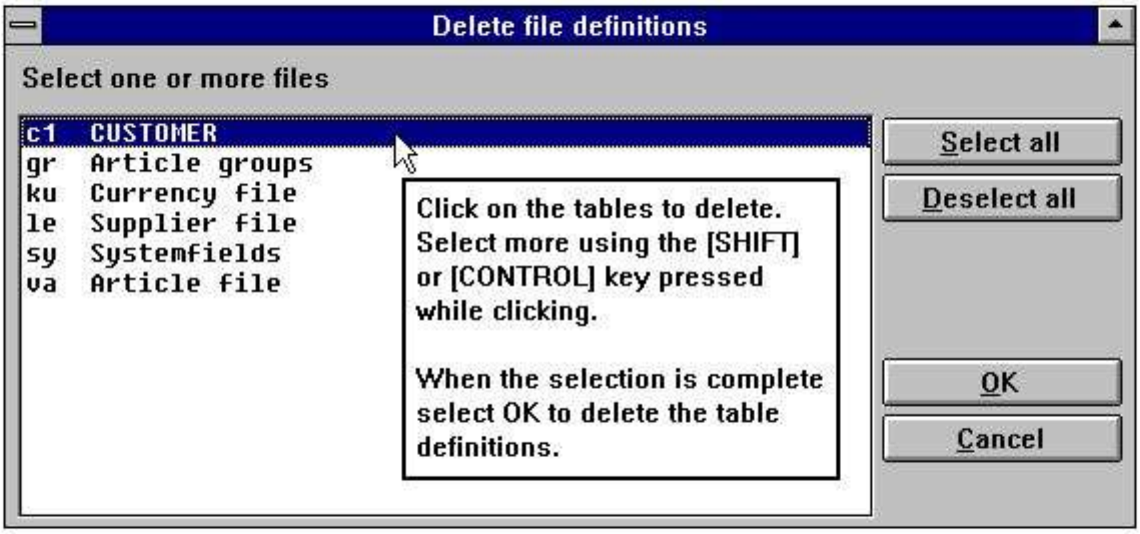
5.2. Deleting tables

Deleting defined tables is done by selecting the following function:

· Select the delete function here



and the following dialog appears:



Select one or more files

c1	CUSTOMER
gr	Article groups
ku	Currency file
le	Supplier file
sy	Systemfields
va	Article file

Click on the tables to delete. Select more using the [SHIFT] or [CONTROL] key pressed while clicking.

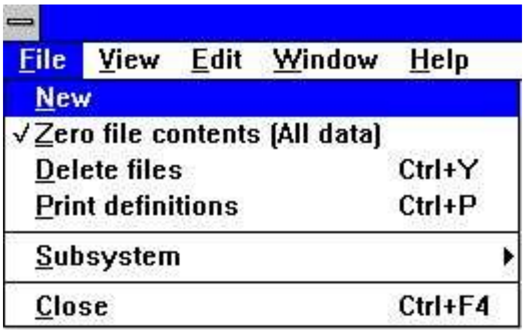
When the selection is complete select OK to delete the table definitions.

35. Deleting tables

Note that in DATAMASTER not only the table definition but also
THE TABLE AND ALL DATA WITHIN THIS is deleted !

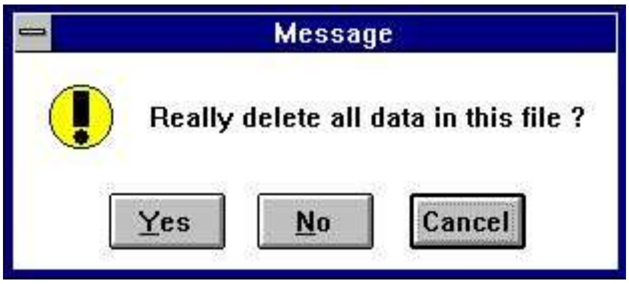
5.2.1. Zero file contents (All data)

This function is normally used to remove test data from the table. The function is activated when you check the line in the file menu:



36. Zero file contents selection

where after you modify a table definition just as normal. When you finishes the modifications you will be asked to confirm:

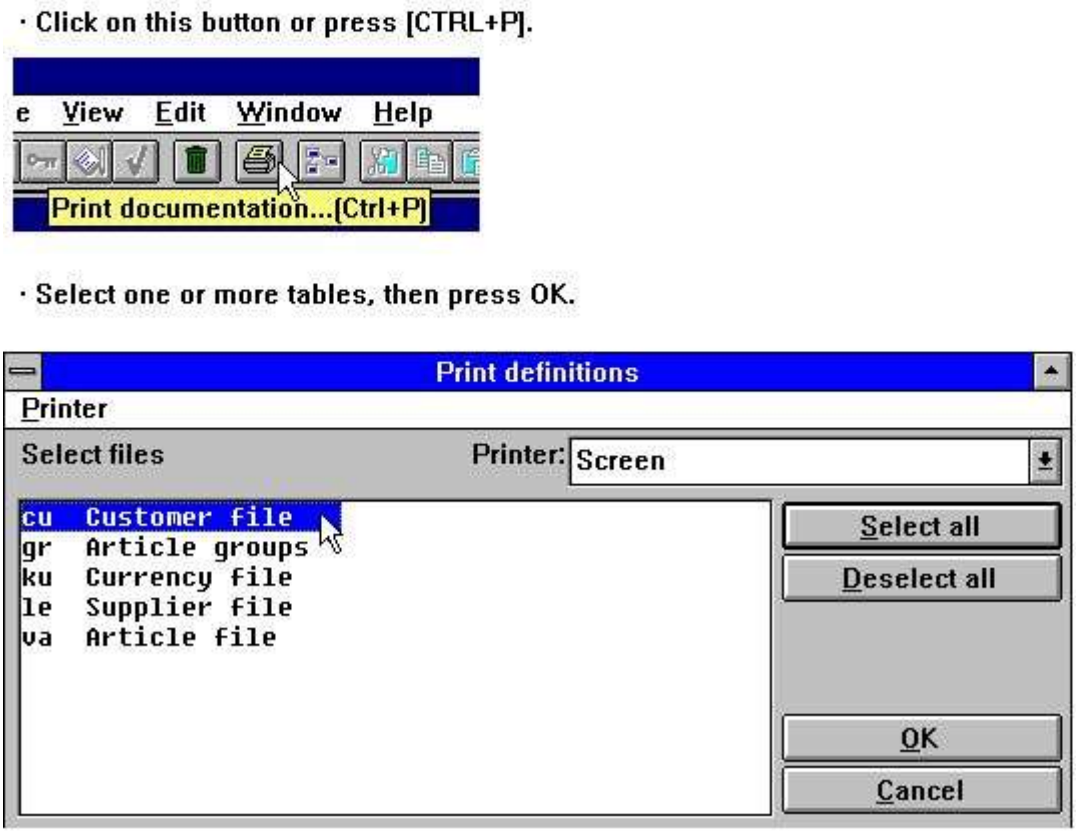

Message

Really delete all data in this file ?

37. Zero file contents confirmation

5.3. Printing documentation

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



· Click on this button or press [CTRL+P].

· Select one or more tables, then press OK.

```
cu  Customer file
gr  Article groups
ku  Currency file
le  Supplier file
va  Article file
```

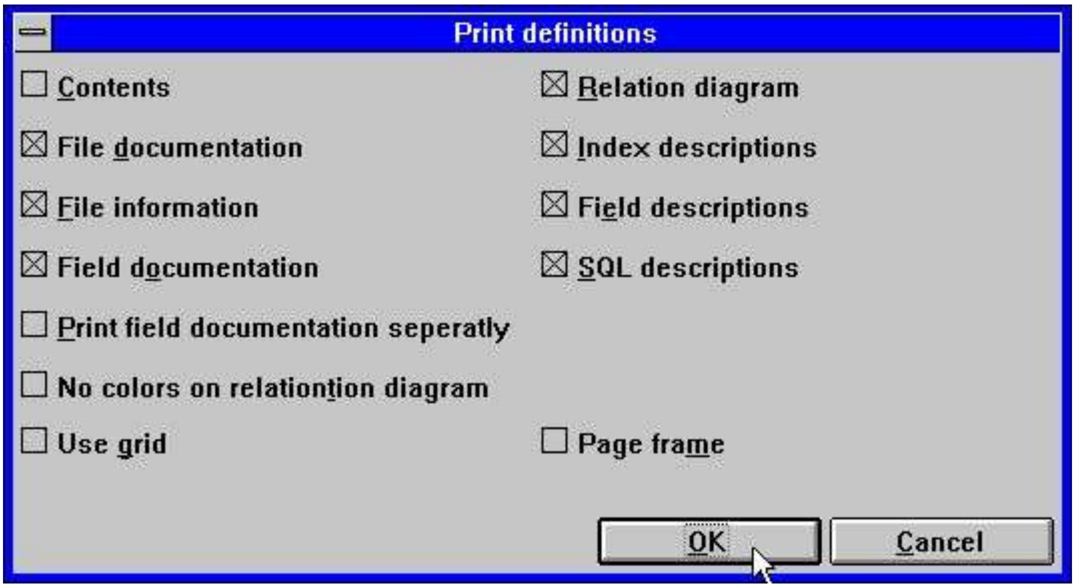
38. Printing documentation

5.3.1. Options

When printing the documentation it is possible to control what to print. A table documentation consists of the following:

- Table documentation
- Table information
- Relation diagram
- Index descriptions
- Field descriptions
- SQL descriptions (printed only for tables using an ODBC driver)
- Field documentation

The options can be switched on or off from the following menu:

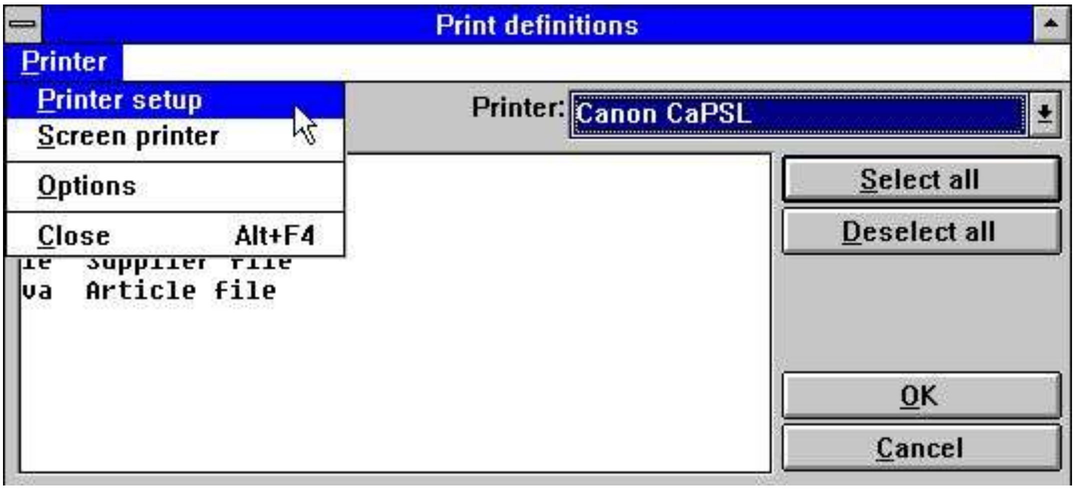


☐ Contents
☒ File documentation
☒ File information
☒ Field documentation
☐ Print field documentation seperatly
☐ No colors on relationtion diagram
☐ Use grid
☒ Relation diagram
☒ Index descriptions
☒ Field descriptions
☒ SQL descriptions
☐ Page frame

39. Documentation options

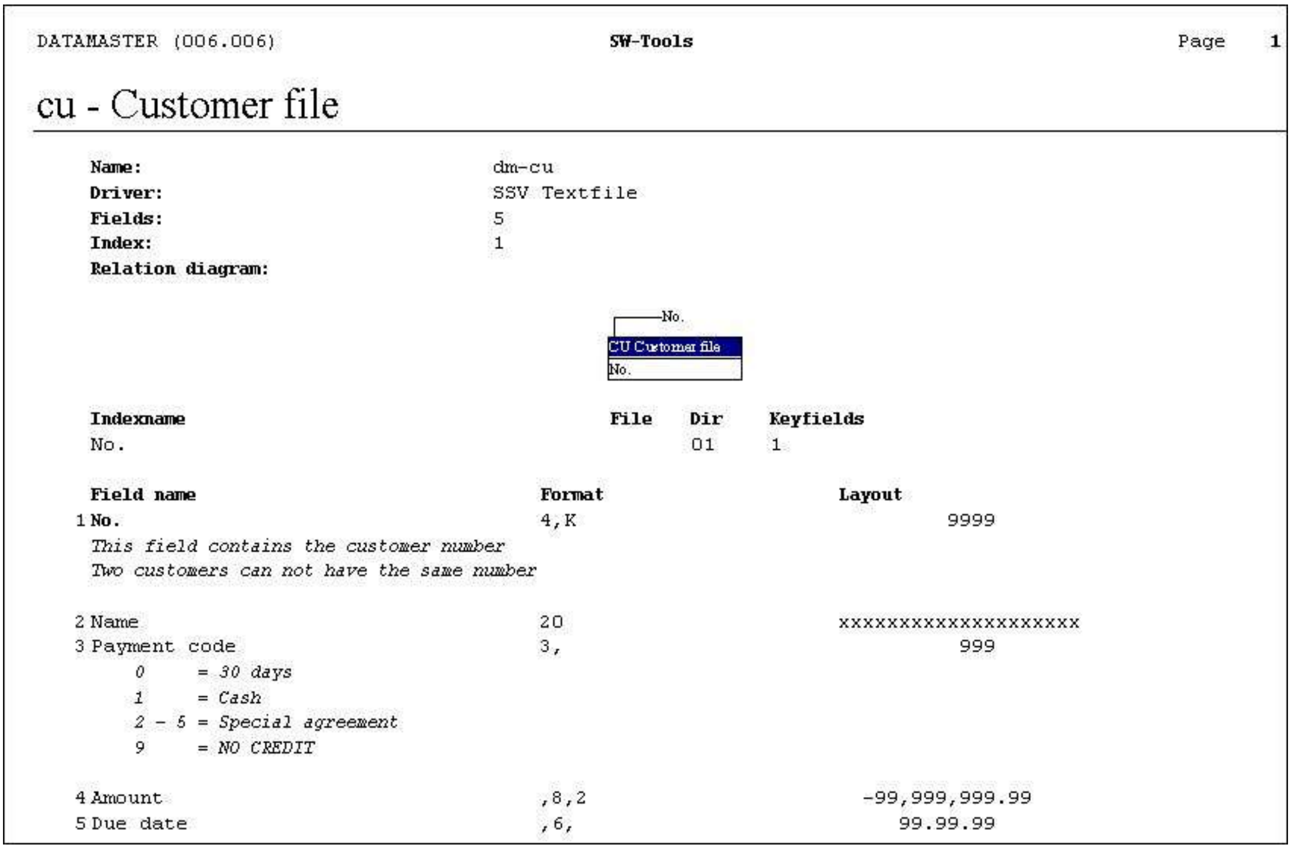
5.3.2. Printer

The documentation can be printed on any Windows installed printer. It can also be printed on a screen printer. In order to select the printer select the following:



40. Selecting printer

This is an example of printed documentation on the table 'cu':



```
DATAMASTER (006.006)                    SW-Tools                                Page    1

cu - Customer file

   Name:                          dm-cu
   Driver:                        SSV Textfile
   Fields:                        5
   Index:                         1
   Relation diagram:
```

No.
CU Customer file
No.

```
   Indexname                                File   Dir   Keyfields
   No.                                             01    1

   Field name                         Format                Layout
  1 No.                               4,K                        9999
   This field contains the customer number
   Two customers can not have the same number

  2 Name                              20                    xxxxxxxxxxxxxxxxxxxx
  3 Payment code                      3,                          999
       0     = 30 days
       1     = Cash
       2 - 5 = Special agreement
       9     = NO CREDIT

  4 Amount                            ,8,2                   -99,999,999.99
  5 Due date                          ,6,                        99.99.99
```

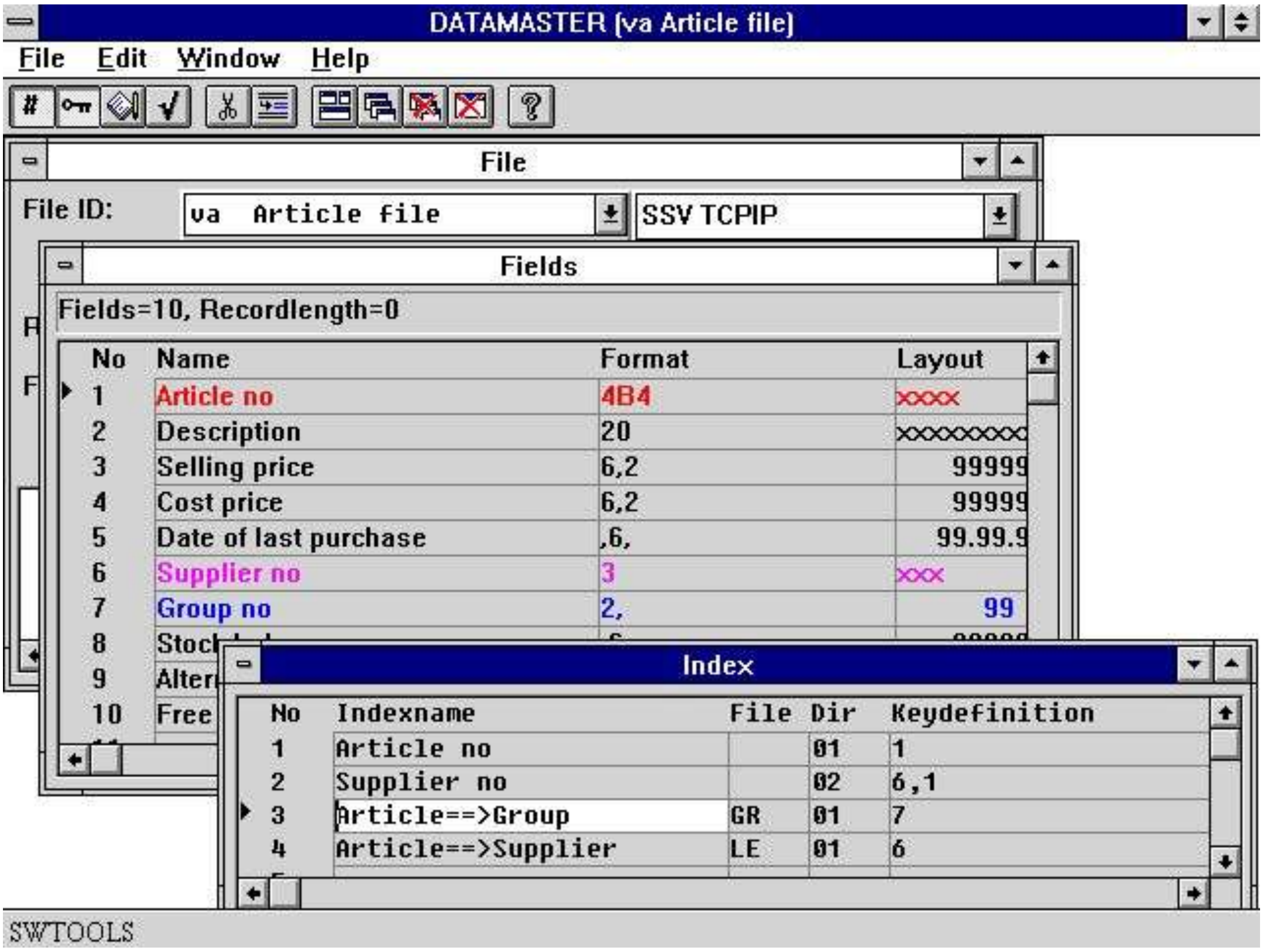
41. Example of table Documentation

6. Working with multiple tables

In DATAMASTER you can structure a complete database system with relations between your different tables. Also relations to non-DATAMASTER files can be established and these files can be mixed with your own files in DATAMASTER, IQ or RAPGEN programs.

6.1. Defining relations between tables

This is done using the INDEX window in the table definition. If we look at the standard article file:



File ID: va Article file | SSV TCPIP

Fields=10, Recordlength=0

No	Name	Format	Layout
1	Article no	4B4	xxxx
2	Description	20	xxxxxxxx
3	Selling price	6,2	99999
4	Cost price	6,2	99999
5	Date of last purchase	,6,	99.99.9
6	Supplier no	3	xxx
7	Group no	2,	99
8	Stoc		
9	Alter		
10	Free		

No	Indexname	File	Dir	Keydefinition
1	Article no		01	1
2	Supplier no		02	6,1
3	Article==>Group	GR	01	7
4	Article==>Supplier	LE	01	6

42. Relations between tables

The two first lines defines real index as article number and supplier number as described above.

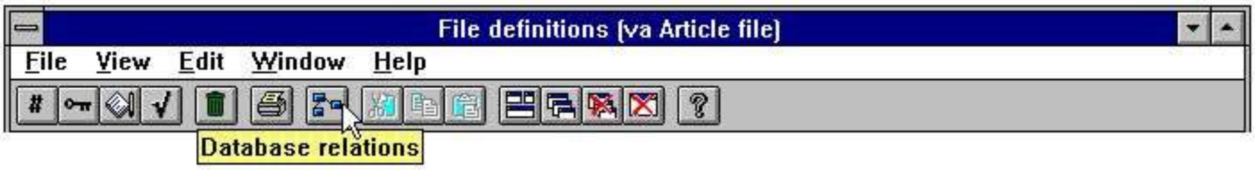
Line 3 defines relation to the GR table using field 7 to search in the index (directory) 1 and read the article group record.

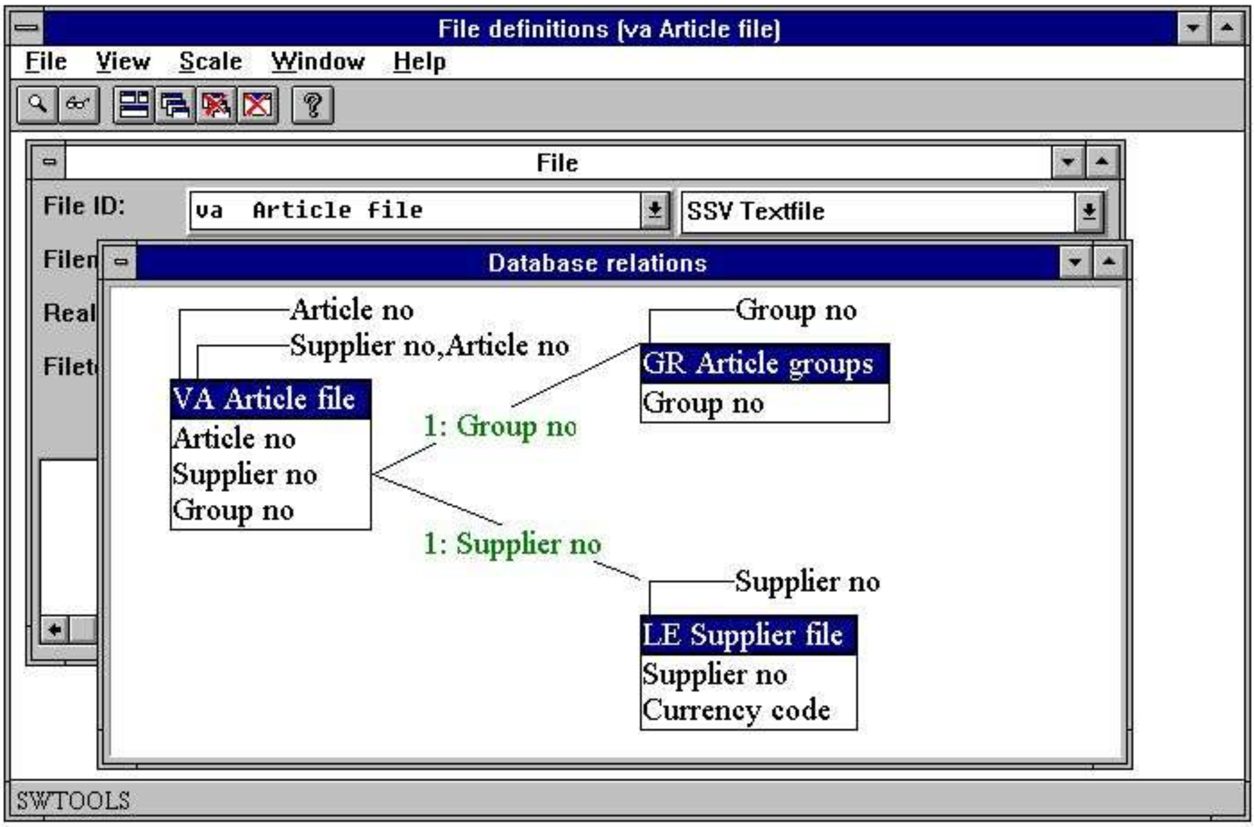
Line 4 defines relation to the LE table using field 6 to search in the index (directory) 1 and read the supplier informations.

Even if you are able to join two tables without having defined a relation between these by stating the relation in READ(xx),..... in you programs we strongly recommend that you define the relations in your Data Dictionary as the use of multiple files then becomes much easier and the documentation will be able to supply you with a relation diagram as shown below.

6.2. Database relations inquiry

When relations between tables exists it can be displayed graphically on screen and printed. To view relations from one table to others select:





File ID: va Article file SSV Textfile

43. Database relations inquiry

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

6.3. Dataentry programs using multiple files

As described in the IQ manual you can make query programs for 2 files just by entering fields as:

va#1-10,le#1-6

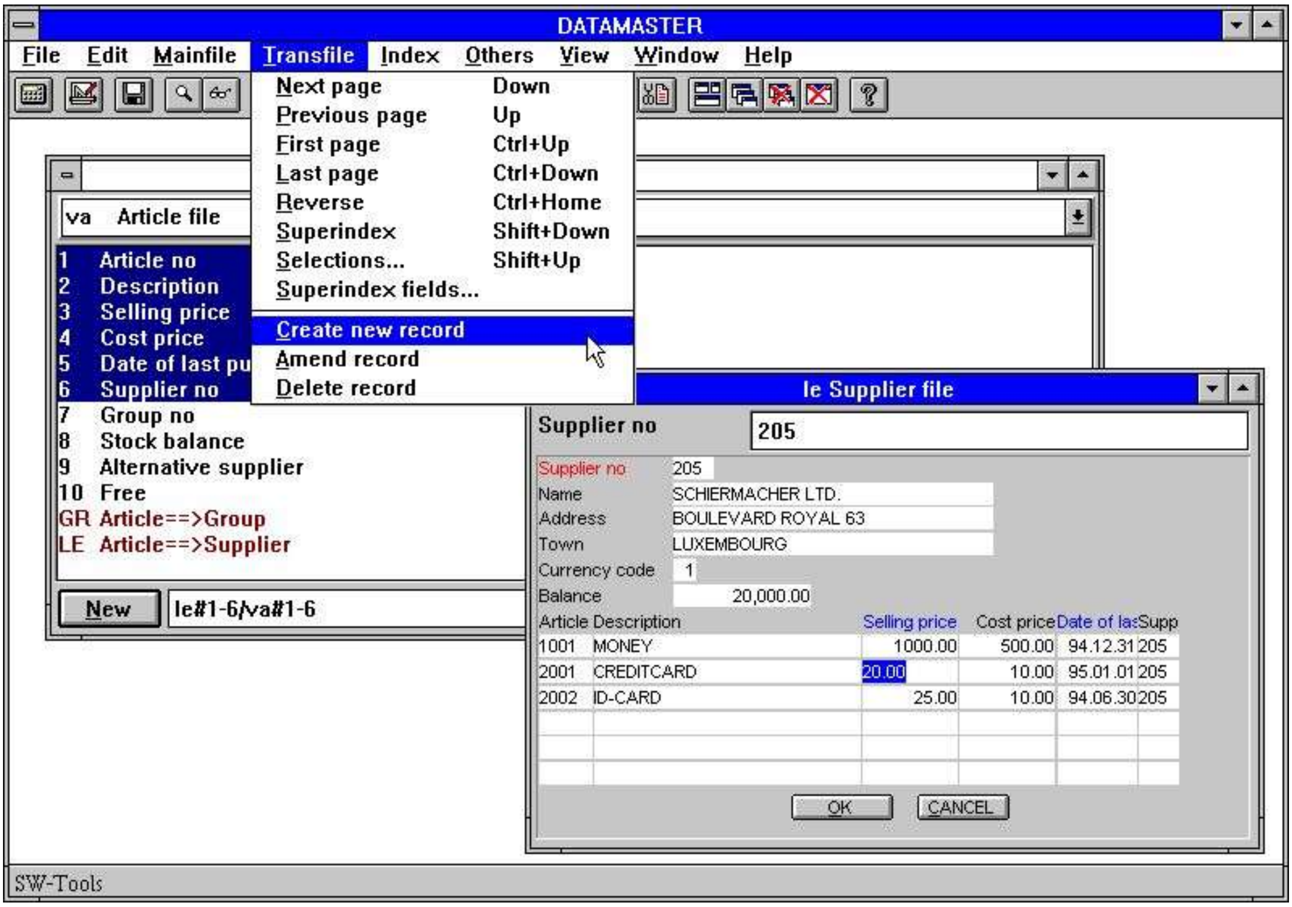
IQ will read the supplier using the defined connection between the two tables. You can connect as many files in this way as you desire.

If the same is made as a DATAMASTER dataentry program you will get a program where you can create/amend/delete records in the main table (va) by standard the secondary table(s), here le, is just being read.

However DATAMASTER makes the dataentry program based on a standard program. When you save this program you are able to amend the calculations placed here for file update and extend these to update not only one file but as many as needed. How to amend calculations is described in the next chapter.

6.4. Transaction registration programs

You may even make transaction programs as shown below:



1 Article no
2 Description
3 Selling price
4 Cost price
5 Date of last pu
6 Supplier no
7 Group no
8 Stock balance
9 Alternative supplier
10 Free
GR Article==>Group
LE Article==>Supplier

le#1-6/va#1-6

Supplier no 205

Supplier no	205
Name	SCHIERMACHER LTD.
Address	BOULEVARD ROYAL 63
Town	LUXEMBOURG
Currency code	1
Balance	20,000.00

Article	Description	Selling price	Cost price	Date of las	Supp
1001	MONEY	1000.00	500.00	94.12.31	205
2001	CREDITCARD	20.00	10.00	95.01.01	205
2002	ID-CARD	25.00	10.00	94.06.30	205

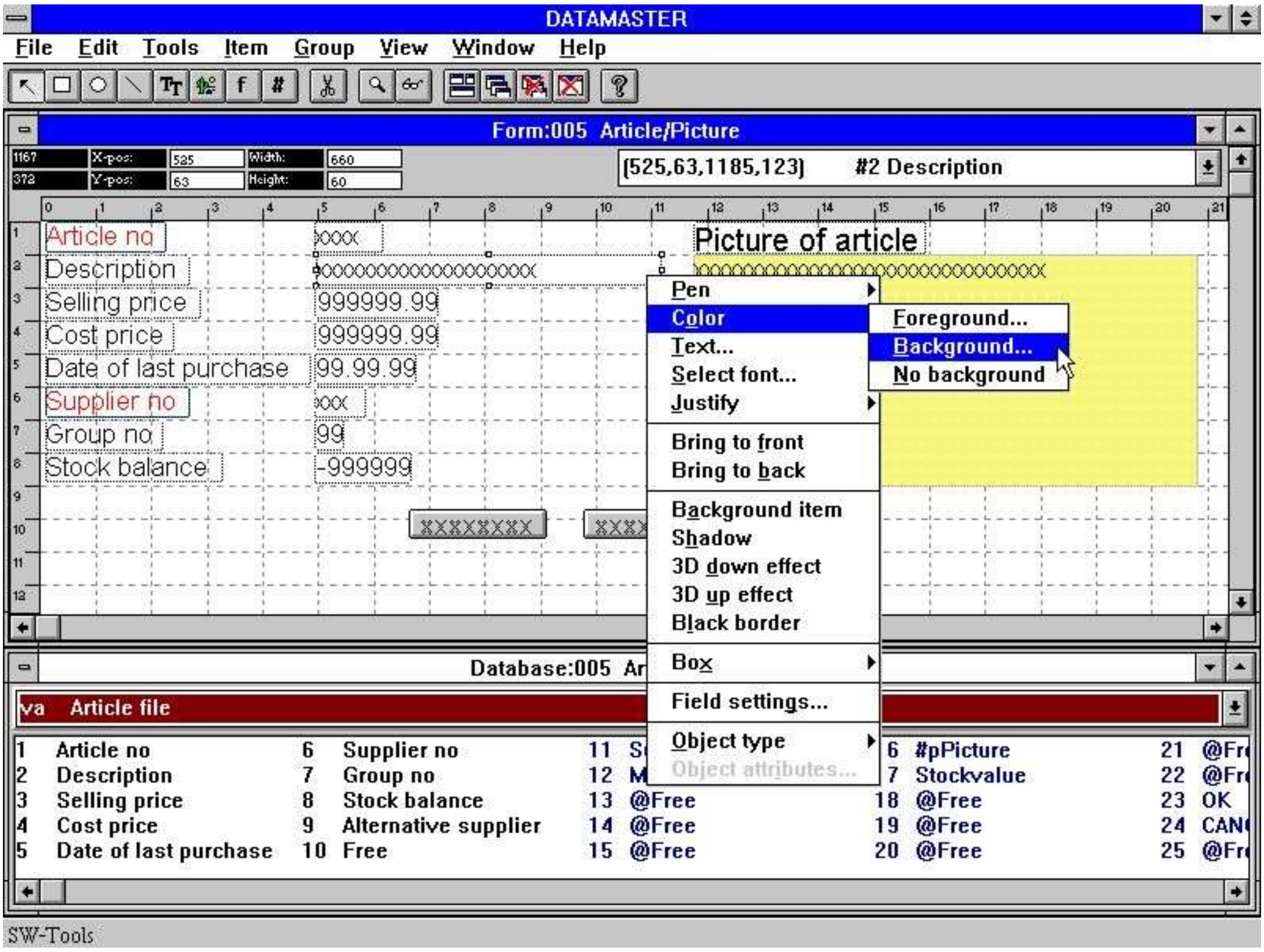
44. Transaction registration

In such program you will have the DATAMASTER functions for insert/amend/delete in both the main and the transaction menu enabling you to create new records in the main file and to insert/modify transaction lines also.

The function of the insert/amend/delete buttons on the toolbar will vary between main and transaction file functions dependent on which field you have currently selected.

7. Saving and modifying the dataentry program

Just like in IQ you can save up to 999 DATAMASTER programs. The menu used for DATAMASTER and IQ will in fact be the same which means you can build up a registration / query system with interaction between different program types. A saved program can be selected from either DATAMASTER or IQ - the program will work in the same way as dataentry or just query.
Modifications of the screen layout follows the same rules as in IQ and you can use all the fancy functions described in the IQ manual for setting colors, drawing circle's and placing pictures on the form:



1	Article no	6	Supplier no	11	S	16	#pPicture	21	@Fr
2	Description	7	Group no	12	M	17	Stockvalue	22	@Fr
3	Selling price	8	Stock balance	13	@Free	18	@Free	23	OK
4	Cost price	9	Alternative supplier	14	@Free	19	@Free	24	CAN
5	Date of last purchase	10	Free	15	@Free	20	@Free	25	@Fr

45. Amending a form

In this chapter we will discuss some principles of calculations in different types of dataentry programs. As we will do file update we will need some more entry points for calculations than for just a query program.
The user interface for calculation input is exactly the same as for IQ - for details on this you should refer to the IQ manual.
The various functions you can use are all described in part 2 - CALCULATIONS and SUBFUNCTIONS.

7.1. OK/CANCEL buttons

When a DATAMASTER program is defined these buttons will be added to ease the use. The OK/CANCEL buttons are examples of the use of DOFUNCTION as these invokes DOFUNCTION(998) and (997) causing the corresponding action to be taken.

7.1.1. Button shortcuts

When you define a button the name of the field is shown inside this. You may enter a freefield name as:

PRESS &ME

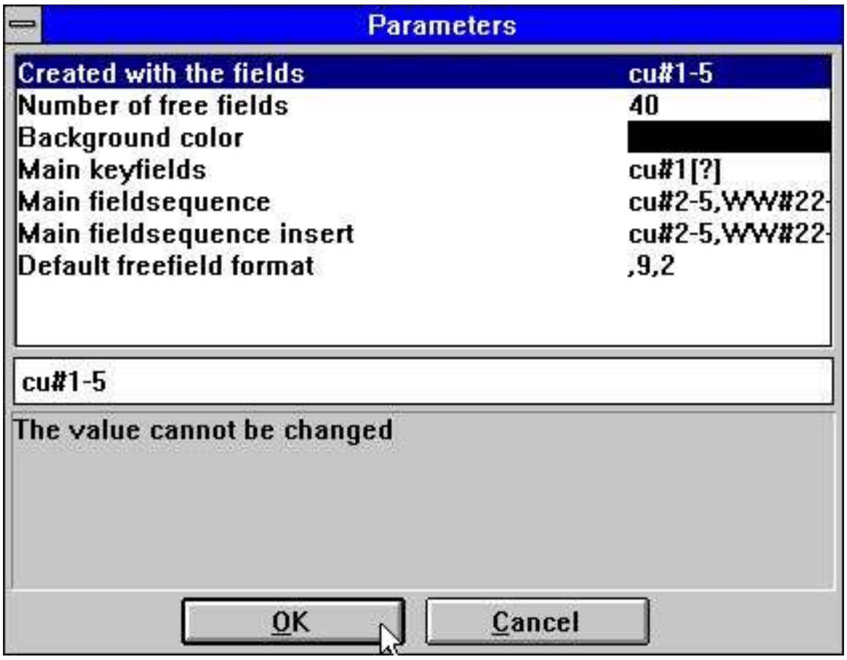
whereas the character just after & becomes a shortcut for this button which then may be activated by ALT+M

7.2. The simple dataentry program

If we define a program like:

cu#1-5

and selects the 'change parameter' function from the file menu we will get the following screen after the program has been saved:



Created with the fields	cu#1-5
Number of free fields	40
Background color	■
Main keyfields	cu#1[?]
Main fieldsequence	cu#2-5,WW#22-
Main fieldsequence insert	cu#2-5,WW#22-
Default freefield format	,9,2

cu#1-5

The value cannot be changed

46. Parameters of program cu#1-5

where we notice some additional DATAMASTER parameters has been placed after the 3 standard IQ parameters fields, free fields and colour.

7.2.1. Field sequence parameters

Here we can amend the input sequence of the fields. The input sequence is used when you just press TAB or ENTER to go to next field, you may always override this by the CURSOR UP/DOWN or by a mouse click or abort from the sequence using ESC or PAGE DOWN.

If a field is not mentioned in the sequence you cannot position to it with a mouse click and you cannot make input to this. The field is displayed only.

There are different field sequences for amending a record and creating a new which DATAMASTER sets equal by default to all used fields except the main key field. By changing these you can define fields which can be entered on create only and not later modified.

The keyfield is kept as a separate field sequence as DATAMASTER needs to know when you are positioned here.

7.2.1.1. Specifying the input sequences

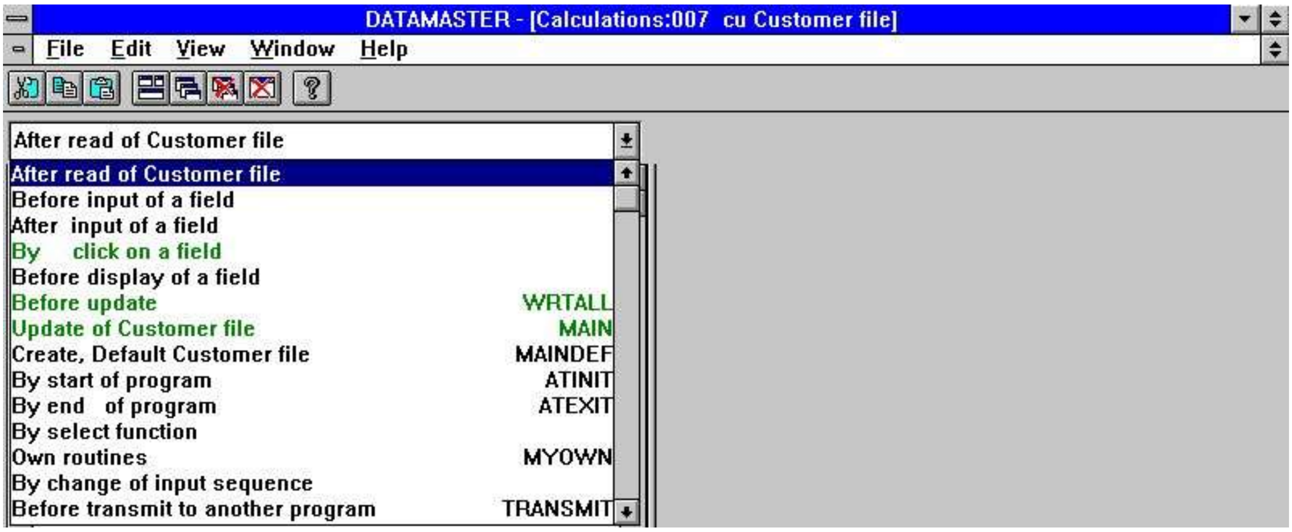
When a new program is builded the sequences will be stated something like:

le#2-6[?]

which means the actual sequence used is le#2-6 the [?] marks that DATAMASTER is allowed to update this if you add new fields to the screen layout. If you want to control the sequence yourself you should remove the [?] and just state le#2-6

7.2.2. Calculation entry points

When choosing calculations on this saved program we notice that these are separated into different parts which can be selected from the listbox:



After read of Customer file

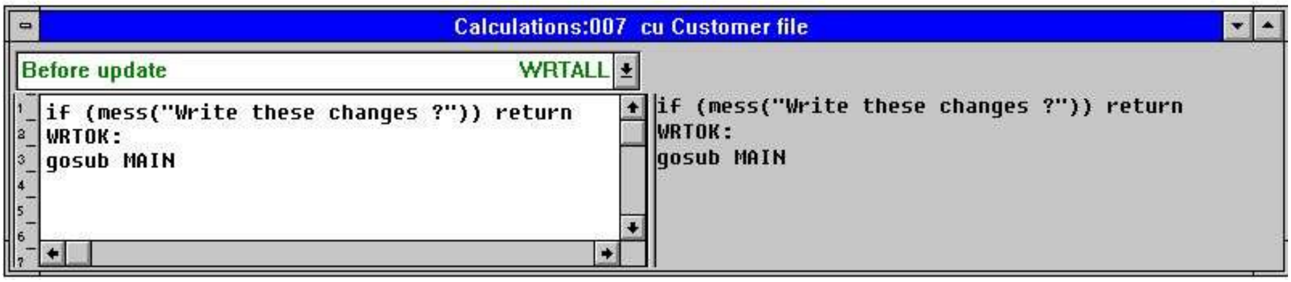
After read of Customer file	
Before input of a field	
After input of a field	
By click on a field	
Before display of a field	
Before update	WRTALL
Update of Customer file	MAIN
Create, Default Customer file	MAINDEF
By start of program	ATINIT
By end of program	ATEXIT
By select function	
Own routines	MYOWN
By change of input sequence	
Before transmit to another program	TRANSMIT

47. Calculation entry points

The lines marked in green in the box contains calculation lines.
By choosing between these entry points we are able to 'stick in' calculations on different places in the DATAMASTER standard program. Let us first have a look at what DATAMASTER itself has placed for us:

7.2.2.1. Before update

The section named 'before update' is activated in the dataentry program when you have made modifications to a record and finishes these.



```
if (mess("Write these changes ?")) return
WRTOK:
gosub MAIN
```

48. Before update

The purpose of these calculations is to decide if update should be made, ready the record for update and then call the actual write routine.

7.2.2.1.1. The MESS function

In the first line we find the confirmation for write. The MESS function displays a message box with the given text asking you for Y/N. If you answer Y MESS returns 0, N causes MESS to return 1.

As the calculations makes: IF (MESS(...)) RETURN a non-zero value returned from MESS will execute the RETURN from this routine and the next lines are not executed. It is answering N skips the oncoming update of the files.

If you delete this line you will get a program where you do not have to confirm writing of each record.

7.2.2.1.2. The GOSUB statement

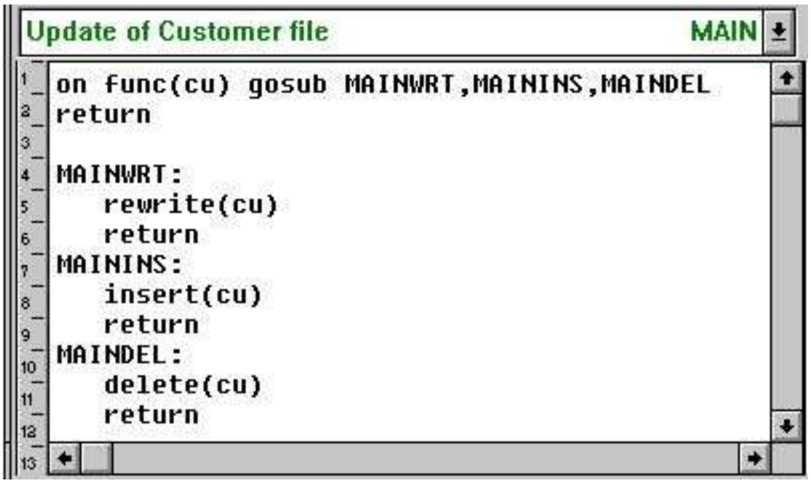
The next line contains GOSUB MAIN which activates the function named MAIN. Each entry point in the calculations has a name. What you see on screen as 'update of customer file' is named MAIN in the calculations.
You can define your own names or labels also by a line like

MYNAME:

in the calculations and use GOTO/GOSUB for this routine also.
If you delete the GOSUB MAIN line you will get a query program, DATAMASTER does not make any updates besides the calculations.

7.2.2.2. Update of main file

This is the real writing of the modified data. Dependent on what you selected in the dataentry program (insert/amend/delete) the record will be new inserted, modified or deleted.



```
on func(cu) gosub MAINWRT,MAININS,MAINDEL
return

MAINWRT:
   rewrite(cu)
   return
MAININS:
   insert(cu)
   return
MAINDEL:
   delete(cu)
   return
```

49. The update itself

7.2.2.2.1. The FUNC function

FUNC(fileid) returns 0,1,2 or 3 giving the selected function of this file:

0 No update needed
1 Amend existing record
2 Insert new record
3 Delete record

7.2.2.2.2. The ON statement

The ON FUNC(cu) GOSUB activates different routines dependent on the value of FUNC, 1 causes MAINWRT to be used, 2 MAININS and 3 MAINDEL.

7.2.2.2.3. REWRITE, INSERT and DELETE functions

The functions makes the physical update of the file. In the standard program only the main file is written but you may extend these routines with write of other files also.

7.2.2.3. Create, Default record

When you choose to create a new record these calculations are done before you enter your data. The block is by standard left blank.

Here you may set default values as #3=7 or even read a complete default record from the file.

7.2.2.4. Calculations after input

For each field you can make calculations which are executed after you have made input to the field. These may make additional validation of the field or update other fields on screen/in the file.

7.2.2.4.1. The DISP function

If you after input modifies any other field on screen than the current input field you will have to place a DISP() line in the calculation causing the display to be updated with the new value(s).

7.2.2.5. Calculations before input

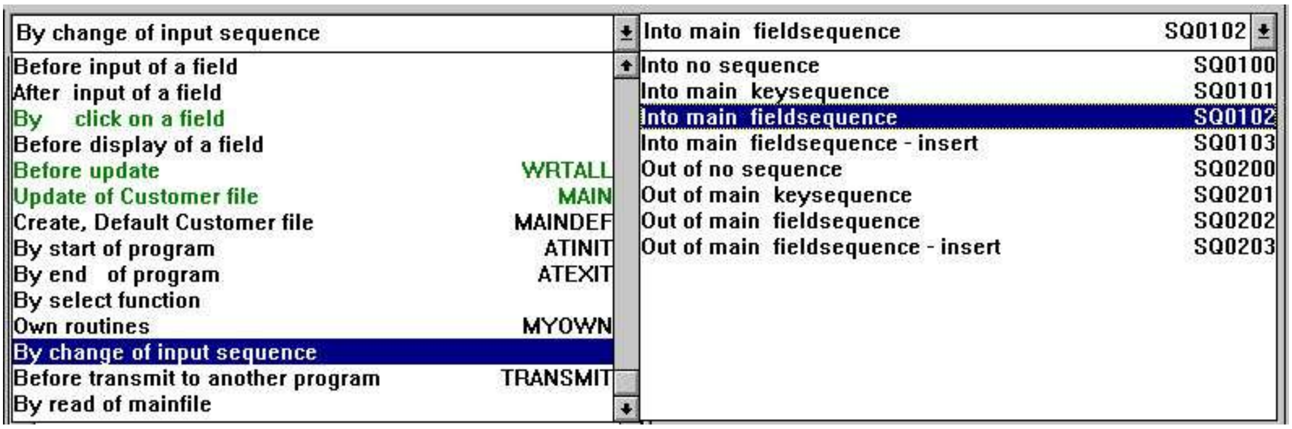
You may also define calculations which are done when you select the field for input but before the input is actually done.

7.2.2.5.1. The NEXTFLD function

You may use NEXTFLD("#3") to jump to field 3 when following the standard input sequence. In this way one or more fields may be skipped or completely blocked out dependent on calculations both before and after the input.

7.2.2.6. By change of input sequence

The PARAMETERS page defines input sequences and whenever the user enters or leaves such sequence a calculation is invoked



By change of input sequence

Before input of a field	
After input of a field	
By click on a field	
Before display of a field	
Before update	WRTALL
Update of Customer file	MAIN
Create, Default Customer file	MAINDEF
By start of program	ATINIT
By end of program	ATEXIT
By select function	
Own routines	MYOWN
By change of input sequence	
Before transmit to another program	TRANSMIT
By read of mainfile	

Into main fieldsequence SQ0102

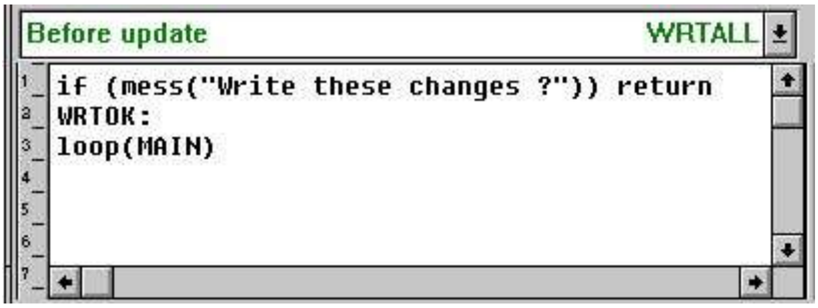
Into no sequence	SQ0100
Into main keysequence	SQ0101
Into main fieldsequence	SQ0102
Into main fieldsequence - insert	SQ0103
Out of no sequence	SQ0200
Out of main keysequence	SQ0201
Out of main fieldsequence	SQ0202
Out of main fieldsequence - insert	SQ0203

50. Calculations by change of sequence

These may e.g. be used to check/read a file when the key sequence is left.

7.3. The list dataentry program

The DATAMASTER parameters and calculation entry points are the same for a dataentry program on list form (cu#1-5l) the only difference is in the calculation for 'before update' which makes a loop over all lines on screen.



```
if (mess("Write these changes ?")) return
WRTOK:
loop(MAIN)
```

51. The loop update in a list program

7.3.1. The LOOP function

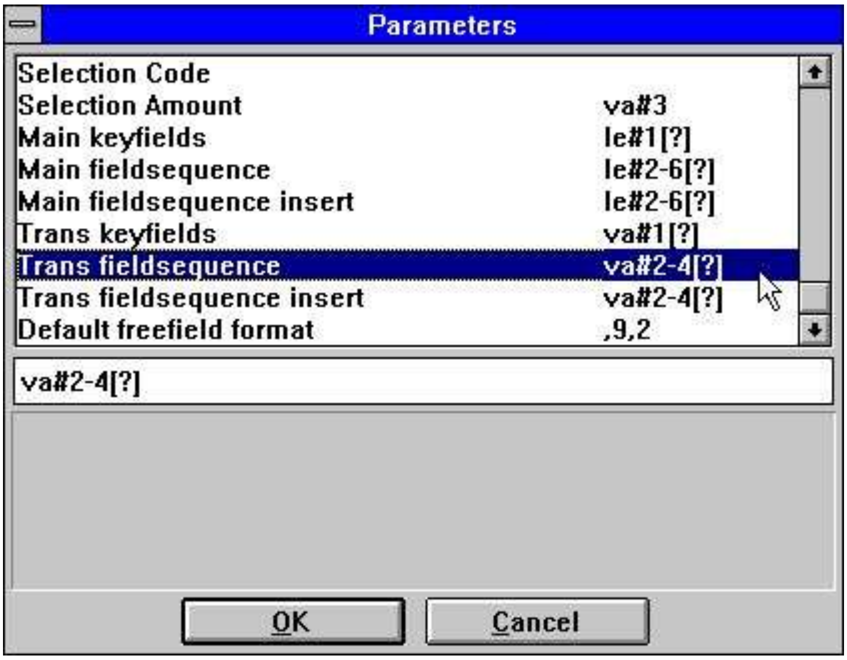
The LOOP function is the same as a GOSUB but is executed not only for one record but once for each line in the DATAMASTER line buffer. The MAIN function then decides for each record, dependent on FUNC, what must be done.

7.4. The transaction registration program

If we define a program like:

le#1-6/va#1-4

and selects the 'change parameter' function from the file menu we will get the following screen after the program has been saved:



Selection Code	
Selection Amount	va#3
Main keyfields	le#1[?]
Main fieldsequence	le#2-6[?]
Main fieldsequence insert	le#2-6[?]
Trans keyfields	va#1[?]
Trans fieldsequence	va#2-4[?]
Trans fieldsequence insert	va#2-4[?]
Default freefield format	,9,2

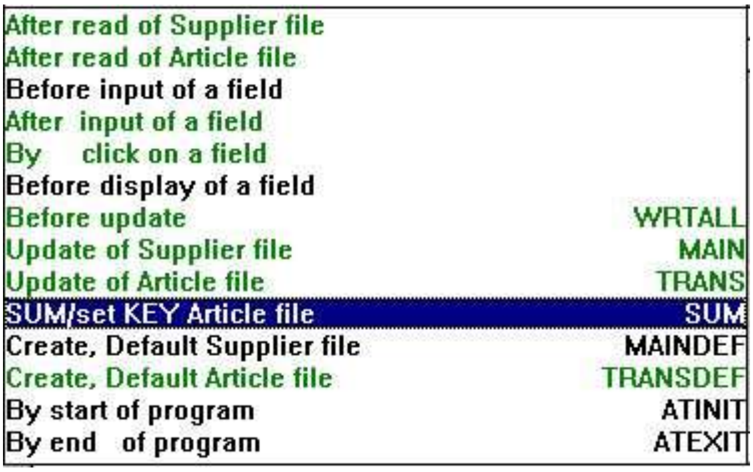
va#2-4[?]

52. Parameters for the transaction dataentry program

We notice that not only the field input sequence for the main file can be altered but also for each of the transaction lines.

7.4.1. Calculation entry points

When choosing calculations on this saved program we get not only the entry points as described above, the UPDATE and DEFAULT has been spilt into main/transaction file and a SUM routine has been added:

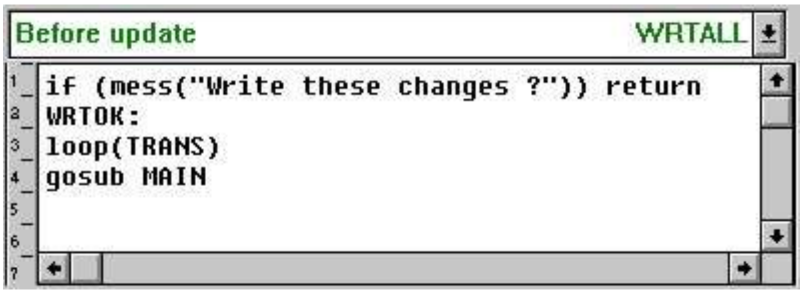


After read of Supplier file	
After read of Article file	
Before input of a field	
After input of a field	
By click on a field	
Before display of a field	
Before update	WRTALL
Update of Supplier file	MAIN
Update of Article file	TRANS
SUM/set KEY Article file	SUM
Create, Default Supplier file	MAINDEF
Create, Default Article file	TRANSDEF
By start of program	ATINIT
By end of program	ATEXIT

53. Entry points for the transaction program

7.4.1.1. Before update

This function now performs a mix of what we have seen in the list and the simple dataentry program:



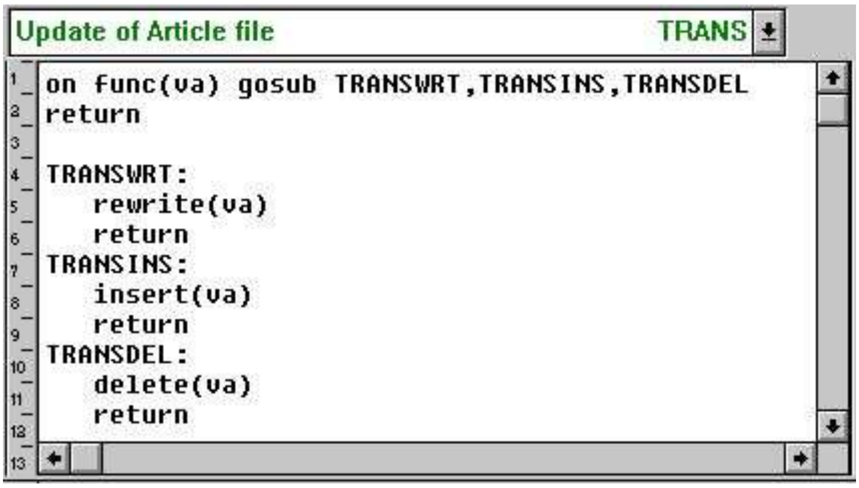
```
if (mess("Write these changes ?")) return
WRTOK:
loop(TRANS)
gosub MAIN
```

54. The transaction loop update

First a loop is done over all transaction in the transaction buffer updating these and then the main file is updated.

7.4.1.2. Transaction update

This routine named TRANS makes exactly the same as the main file update but is called for each transaction line.



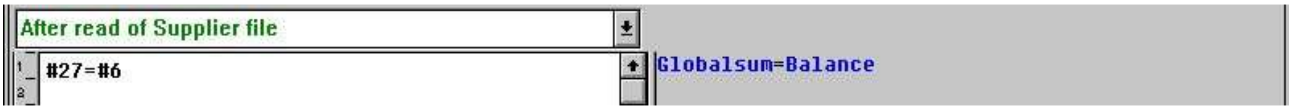
```
on func(va) gosub TRANSWRT,TRANSINS,TRANSDEL
return

TRANSWRT:
   rewrite(va)
   return
TRANSINS:
   insert(va)
   return
TRANSDEL:
   delete(va)
   return
```

55. The transaction update

As for the main file update routine you may extend this to update more files.

7.4.2. SUM of transactions

On a transaction query program like this IQ will generate SUM routines to totalize the transactions when these are being read from the file:

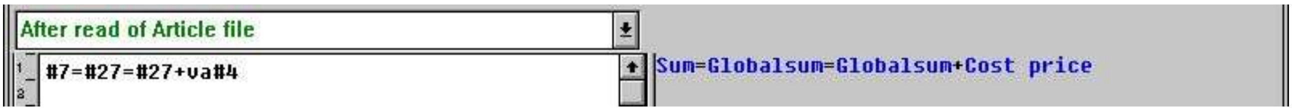


After read of Supplier file

```
#27=#6
```

```
Globalsum=Balance
```

56. IQ Initializing SUM with brought forward amount



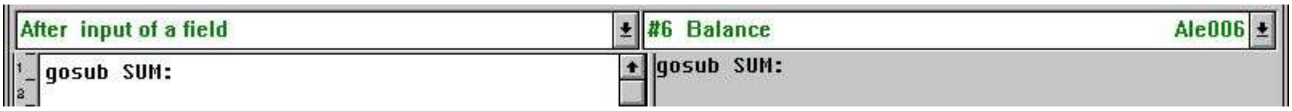
After read of Article file

```
#7=#27=#27+va#4
```

```
Sum=Globalsum=Globalsum+Cost price
```

57. IQ adding up SUM

In a DATAMASTER dataentry program you may amend the fields from which this sum is calculated. This must then cause a new calculation of the sum why DATAMASTER also places:



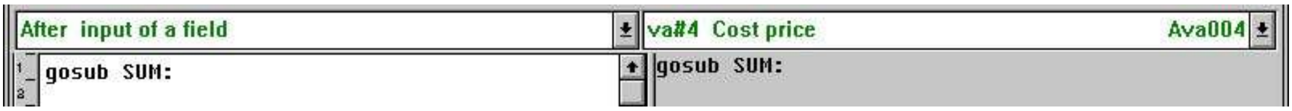
After input of a field

```
gosub SUM:
```

#6 Balance Ale006

```
gosub SUM:
```

58. Calculation when changing the brought forward field



After input of a field

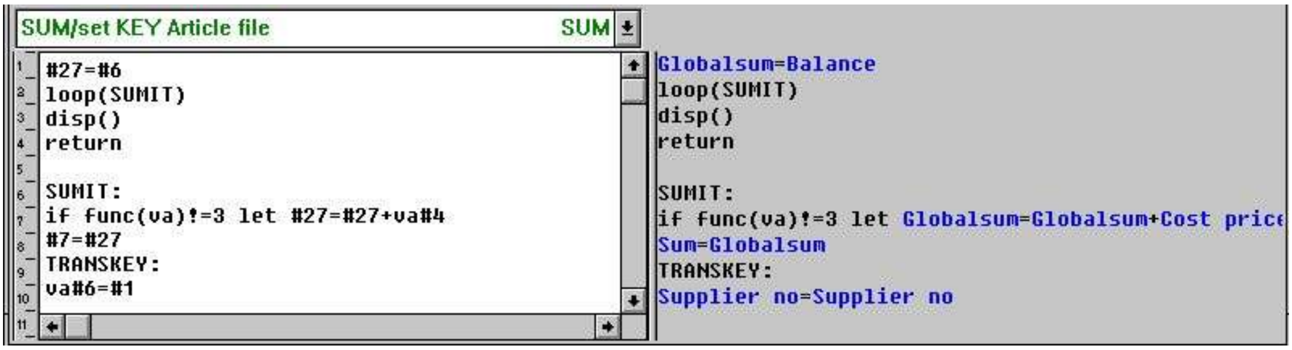
```
gosub SUM:
```

va#4 Cost price Ava004

```
gosub SUM:
```

59. Calculation when changing the transaction amount

with reference to a SUM routine recalculating the sum all over:



SUM/set KEY Article file SUM

```
#27=#6
loop(SUMIT)
disp()
return

SUMIT:
if func(va)!=3 let #27=#27+va#4
#7=#27
TRANSKEY:
va#6=#1
```

```
Globalsum=Balance
loop(SUMIT)
disp()
return

SUMIT:
if func(va)!=3 let Globalsum=Globalsum+Cost pric
Sum=Globalsum
TRANSKEY:
Supplier no=Supplier no
```

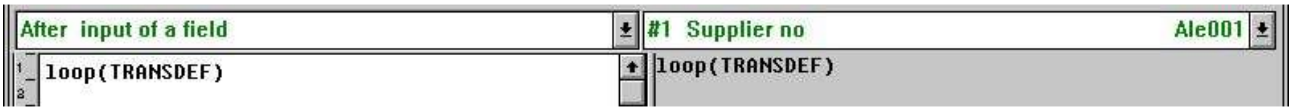
60. The SUM routine calculating transaction total

This SUM routine consist of 2 parts, the first part initialising the sum and looping over all transactions, the 2 part actually calculating the sum for each transaction. The IF FUNC.. ensures that deleted transaction lines are not taken into the total.

The last 2 lines labelled TRANSKEY: ensures the transaction key matches the main file key and is used in the following.

7.4.3. Set KEY on transaction file (SUM)

In a transaction dataentry program we normally have some 'critical' fields in the main file. A change of such a field must cause all transactions to be updated. We have looked at the 'brought forward' field already (le#6) which triggers the recalculation of SUM.
The connection between the main file and the transactions are based on one or more key fields held in the transaction matching the main file key. If the main file key is changed all transactions should be moved also.
To do this DATAMASTER makes a calculation triggered when you changes any field in the main file key:



After input of a field

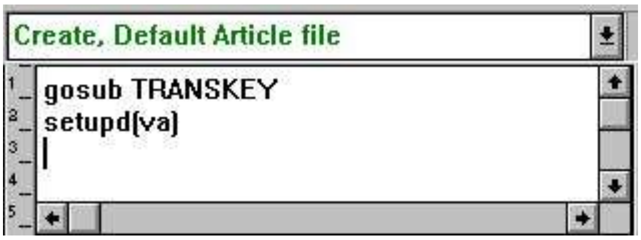
```
loop(TRANSDEF)
```

#1 Supplier no Ale001

```
loop(TRANSDEF)
```

61. Calculation when changing the supplier number

calling the routine TRANSDEF for each transaction setting the key:



Create, Default Article file

```
gosub TRANSKEY
setupd(va)
```

62. Calculation when creating a new transaction

7.4.3.1. The SETUPD function

In the calculations for update of the files we made a check on FUNC whether a specific record was changed.
SETUPD(xx) marks a line as modified and is used when the main file key is modified to tell all transaction lines must now be written.

7.5. Other calculation functions

Some other functions specific for DATAMASTER should be mentioned here:

7.5.1. The SEQ function

The SEQ function can be used to change the input sequence. Placed in a calculation it changes the value of the parameter given on the parameter page.

7.5.2. The MENUS function

This function can be used to deactivate menus which means you can make programs that can only amend records the insert/delete functions being blocked.

Figure list

Index