SWT
SOFTWARE TOOLS

Dansk Brugerhåndbog

Duevej 23
DK-3250 Solrød Strand
Denmark
Phone: **+45) 33 33 05 56**
Mail: **swtools@swtools.com**
www: **www.swtools.com**

TRIO Frigivelses information (008.000)

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

Indholdsfortegnelse

1. Forord

The release 008 of TRIO contains the following enhancements:

Users
- User maintenance and login (also without user administration)
- Report start statistics (also without user administration)

.

Print
- PIP Print archiving
- Formulary print on reports
- Support of different picture formats (GIF/TIF/PCX etc)

.

Runtime
- ZIP and UNZIP functions for packed file handling
- RUNTIME package and license handling

.

IQ
- GRID listboxes
- DGRID data selection input listboxes
- Scrollbar type objects

.

Datadictionary
- Search facilities on files and fields
- Extension of file ID from 2 to 8 characters
- Extension of fieldnumbers from maximum 999 to 999999999
- Raising of different limits (fieldlength/recordlength etc)
- Integration of different database definitions in one program
- Datadictionary calculations
- Datadictionary standard field formats (like AMOUNT and DATE)

.

Languages and installation
- Language support of French, Spanish, Swedish
- Installation program updated to 32 bits long pathnames

.

Manuals
- Now online on the CD together with a print program

.

For customers with maintenance contract the additional functionality will be free of charge, these customers will receive an update when the test period has ended.

2. Start and installation of TRIO

2.1. Start of TRIO

The startscreen for TRIO has been changed to look as:

1. Start of TRIO

2.1.1. User login

TRIO without user administration has been opened to start with a username anyway in order to take advantage of the HOMEPATH, which may be given for each user.

2. Start of TRIO when more users are activated

2.1.2. Logo

You may design your own logo, store this in the file SWTRIOLL.BMP which will display like:

3. Start of TRIO with own logo activated

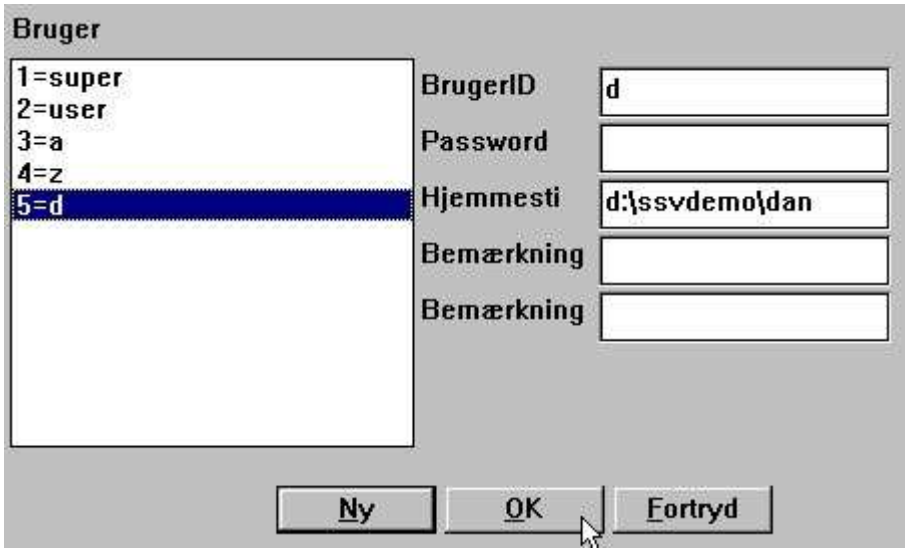
2.1.3. User maintenance

Users may be defined even if the user administration is not activated, however no individual user privileges can be made.
This means the User administration may be selected and setup, but the definition screen is without the privilege definitions:

4. The reduced user menu

2.1.3.1. Creating the users

On the user maintenance screen the fields for user group and grant/revoke are omitted as all users becomes superusers when the user administration license is not present.



5. The reduced user maintenance screen

Apart from these reductions the user maintenance follows the procedure as described in the user administration manual, and you gains the advantage of the HOMEPATH enabling you to split different applications on one system dependent on login instead of using subsystems or different Windows icons.

2.1.3.2. Upgrading with real user administration license

Without the module user administration is present on the license the main user SUPER is created as superuser (usergroup 1-Revoke), all other users as USER (usergroup 2-Grant), but all threaded as superusers with all rights including right to create other users.
If user administration later is added to the license, even without being X-Marked for the specific PC, the treatment of all users as superusers is terminated and the first user (SUPER) only may create other users.
For security reasons, a user password file created with user administration license activated, cannot be used directly on a PC without this license, as this would violate the protection system.

2.2. Installation and upgrading

Due to support of more languages the installation screen has been changed to:

6. The language screen

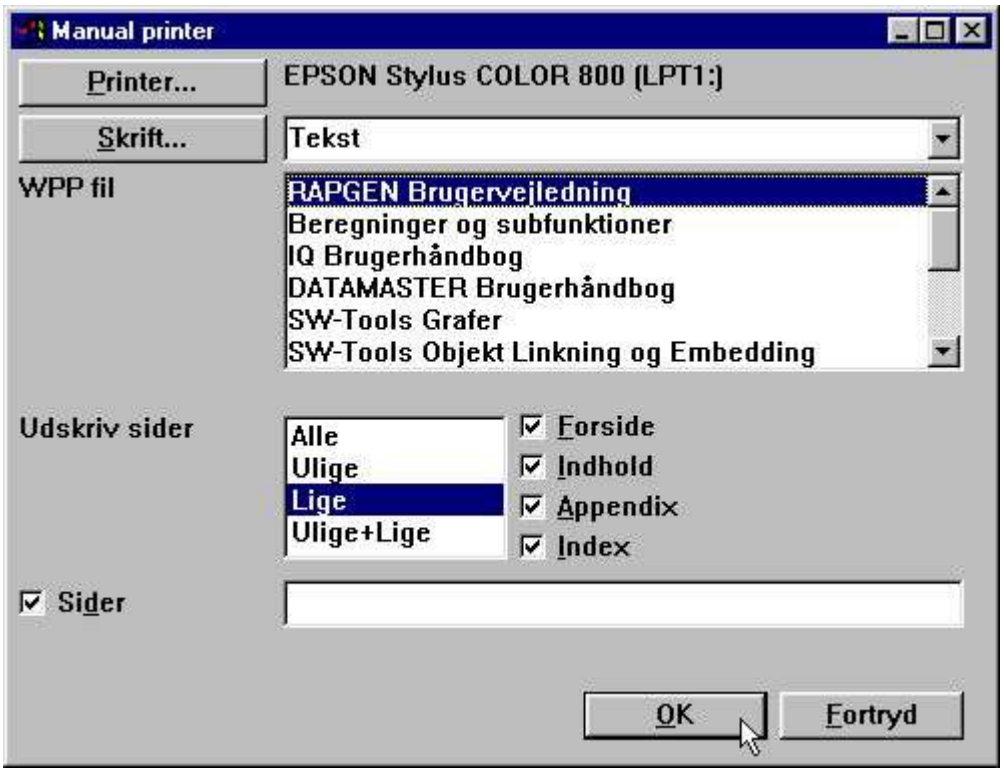
7. The 16/32 bit screen

8. The product screen

The installation program for the 32 bit version is now running 32 bits also, which allows the use of long filenames already by installation.
TRIO version 008 if fully compatible with version 007, but you should note that the opposite is not the case. If you use fileID's of more than 2 characters no backward compatibility is possible, file definitions cannot be exported to previous versions or to the VIEW basic report generator.
Also the use of database interaction, external packages, GRID functions or PIP print collection will not be moveable to TRIO versions below 008.
By upgrading it is possible to use any persistent 007 server module, but we recommend upgrading the server to version 008 also, which will be able to service a mix of clients on newest or older versions. It is not possible to use long fileIDs, fieldnumber modifications or like with a 007 server.

2.3. Manuals

Manuals will from version 008 be distributed on CD only together with a program enabling the user himself to print as many copies as needed:



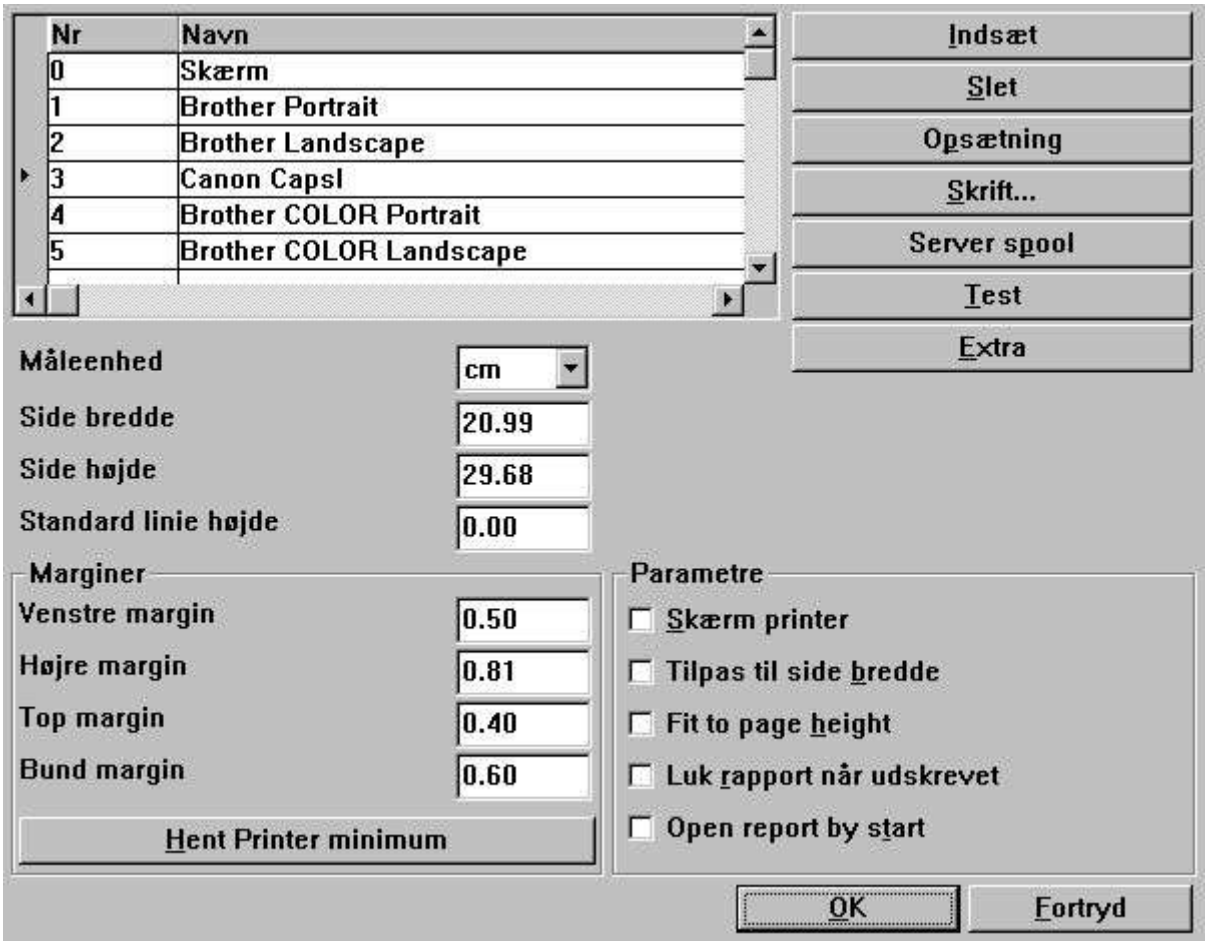
9. Printout of manuals

3. Printing from TRIO

In order to make reports more distributeable in package form, where change of printers is most likely when the package is installed, field boxes in the layout will now be set to automatic resizing by default, as has some of the printroutines been reworked.
However you should note that when inserting a field in a report the selected font may be a PRINTER font, which varies with the printer used. A good advise when developing reports to be used on different printers is to use a general font only as times new roman or like, and to force the printer driver in windows to download TrueType fonts, not using any hardwired fonts in the printer.
Also new facilities for print archiving and form print has been added.

3.1. Printer setup

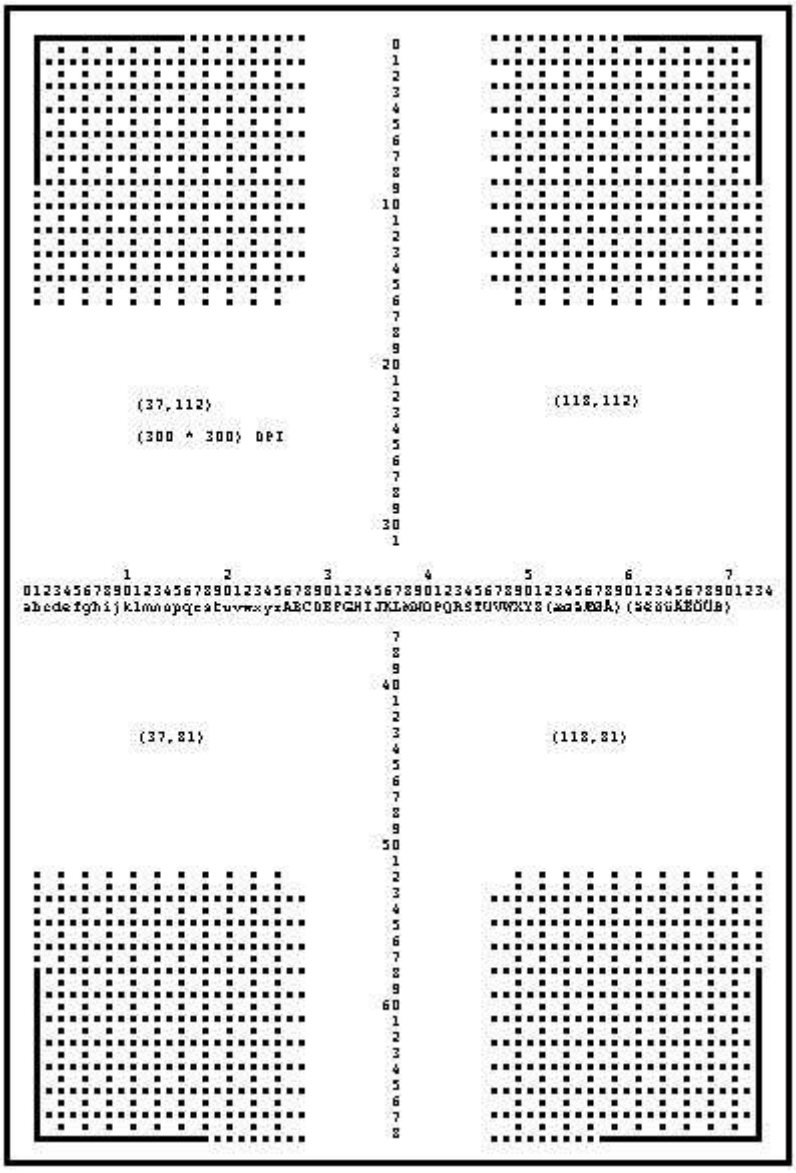
The printer setup has been extended with a TEST and an ADVANCED button:



10. The printer setup

3.1.1. The TEST button

When TEST is pressed one page is printed on the selected printer showing the margins and resolution used.
The corners on this testprint are marked with a grid where the margins should be shown as a solid line. As printers seems to behave very different you should check the following before trusting the driver settings:



11. Test button, CAVE before trusting printer

3.1.2. The ADVANCED Button

The ADVANCED buttons are used to setup the new facilities for form print and print archiving.

12. The advanced button in the printer setup

3.2. Forms

The formula file should contain a picture in WMF format, which then will be extended to the full paper size (without respects of any margins and with stretching the image to fit both directions) and printed transparent behind all other text on all pages.
As an example you may try to enter one of the picture files from the TRIO demo system, which may be stored as

c:\swtools\demo\wmf\0101.wmf

You should note, that the form is output to the printer for each page, this may reduce the speed of printing drastically on some printers.
If possible for production runs, use any macro language for the printer to download the form fixed into the printer memory as described in the individual printer drivers manuals.

3.2.1. Creation of forms

To create a form you may use any picture program able to save in WMF format, Microsoft PowerPoint could be a candidate here.
The reason for using the WMF format is because it is based on vectors, e.g. the drawing may be scaled without loosing its resolution. However, it is also possible to use any of the other supported pictures format, such as BMP, GIF, JPG etc.

3.2.2. Controlling the form print from the calculations

It is also possible to control the form printout from the calculations, that means a report may use different forms on the pages printed.

PRINT(FORM=c:/swtools/demo/wmf/0101.wmf)

will change the form to be taken from the WMF filename given.
The PRINT(FORM= should be placed in the FIRST section if the form is to be used from the very start. If placed in the NORMAL section, the form change will take effect, when the first line on the nextcoming page is printed.

3.2.3. Handling of forms during the layout definition

As soon as a form has been setup for a printer you may select this printer in the layout function which will then show the form in background according to the printer page size.

3.3. PIP Print archiving

Setting a PIP (Print zIP) archive filename causes all printout to this printer to be archived into filename.zip and a printoverview file filename.ssv to be builded. An example may be:

c:\keepit\kept

If the directory c:\keepit is not present, this will be created. Note that the first character of the filename must be a letter. You should omit the file extension as this is set to zip/ssv

If you omit the path from the filename, the database path from the setup preferences will be used.

If the filename is 4 characters only or less, YYMM will be added when archiving to keep filesizes low and ease deletion of old print. The above example will produce the files

c:\keepit\kept9908.zip
c:\keepit\kept9909.zip
c:\keepit\kept.ssv

The print archiving may also be controlled from within the calculations of a report without defining this on the printer setup, see the PIP function later.

Print archiving may be defined for a screen printer also so a physical print is produced only as a reprint from the archive.

3.3.1. Screen printer display when using PIP

When you are using PIP print archiving for a report printout, the VIEW button will be activated even if the output is not printed on a screenprinter, as the screenprinter files have to be made anyway for the print archiving.
That means you get the double function of printing and in the same time viewing the output on screen.

3.3.1.1. Screen printer display without using PIP

You may specify the making of screen printer files for a printer without actually archiving the print by entering the

PIP Filename as just -

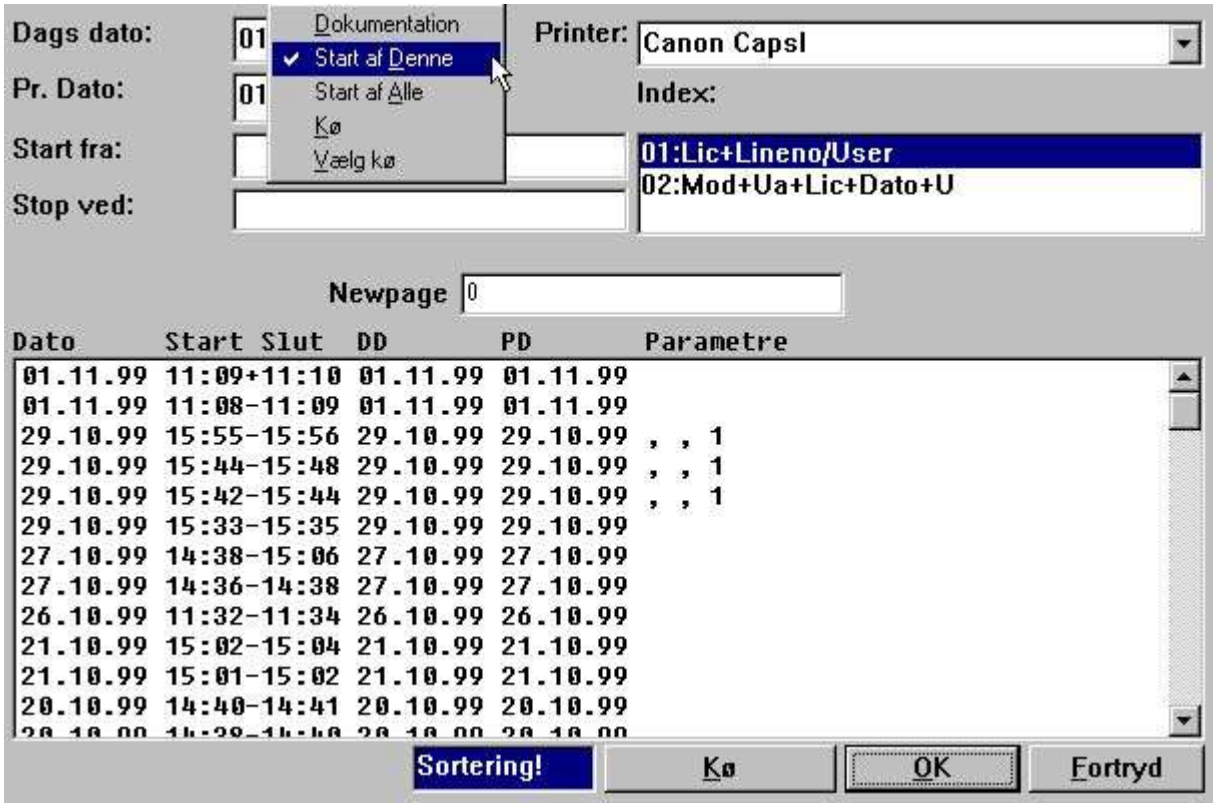
the print is not archived but you get the VIEW possibility.

3.3.2. Recalling an archived print

The report statistics startlog will show start-endtime for a run, if print has been collected the - is replaced by + as start+endtime, by click on the endtime you will get the print on screen and may reprint all or selected pages. The reprint may be done on a different printer type as the print will then be resized to this, but please observe that differences in page setup and field scaling may heavily influence the print.
Note that the + mark indicates only that a print was collected, not necessarily that the print is still present, as somebody may have made a periodically cleanup of the zipfile in which case the print of course cannot be displayed.
As the startlog holds the last 100 starts only the recall possibility will be pushed out after more than this number, as is this recall restricted to report start and not other print such as data dictionary definitions.
However you may define the SSV file made by the print archiving for an IQ program, use this for searching a specific print and display it with the calculation function SCRPRT, see later.

3.3.3. Report start statistics

The report start statistics is now included as a part of TRIO without a licence for user administration has to be present. For a full description you should refer to the last chapter in the user administration manual.



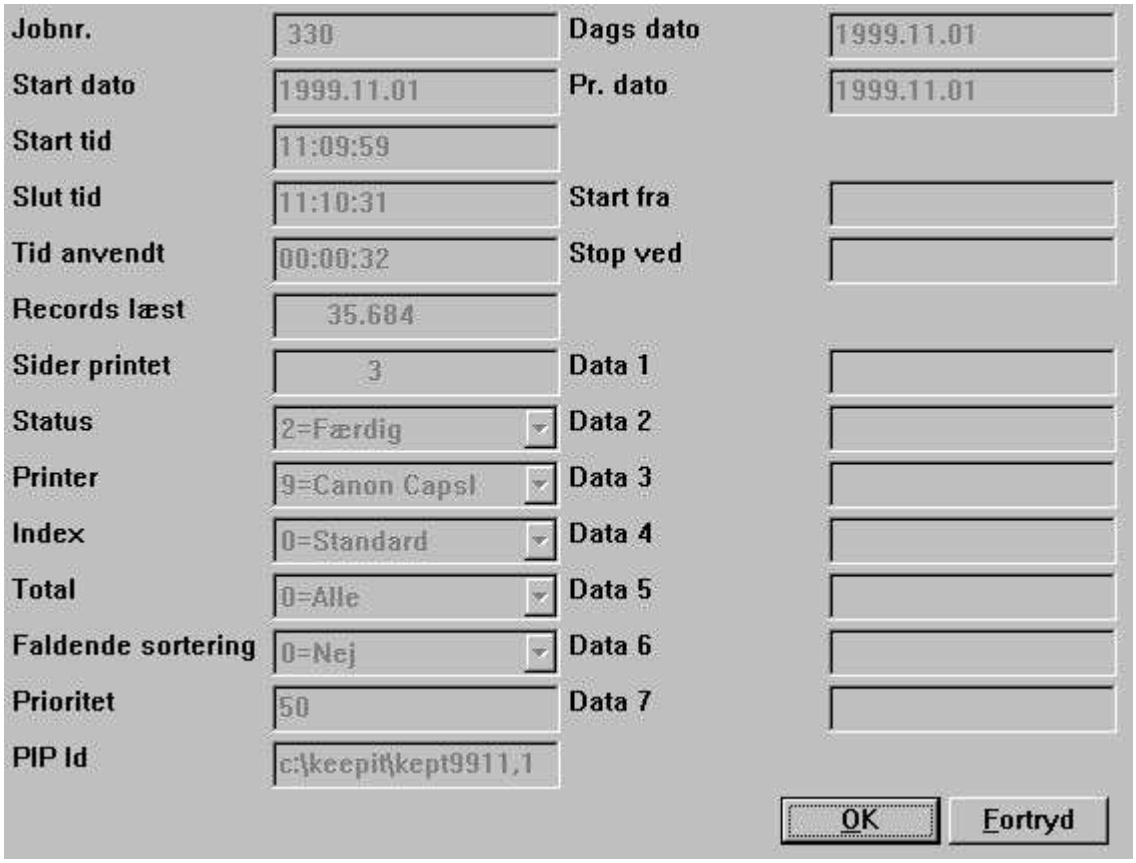
13. Start of a report showing the report statistics

3.3.3.1. Use of the report start statistics

By click on one of the statistic lines different functions is achieved dependent on where the click is made:

- **Startdate gives details of this report**
- **Starttime gives run statistics for this start**
- **Endtime displays the archived print if present (marked with +)**
- **Anywhere else duplicates the parameters with/without the dates**

The PIP print archive file and number will be shown also on the report detail information you will get by clicking on the date field on the line:



14. Display of report details with the PIP id

3.3.4. Structure of the PIP files

The print will be stored in the same way as for the screen printers output in one or more compressed ZIP-files.
Each print session is numbered taking the next free number from the zipfile, the printsession number nnnnn may range from 00000 to 32000.
Each printpage will be stored as a file nnnnn-NN.NNN in the zipfile, nnnnn being the printsession number, NN.NNN being the print page number within the printsession. These filenames are 16/32 bit compatible.
Any chart on the report will be included in the print archiving also, but you must note, that pictures and OLE objects may be stored as references to an external file, not packed into the print archive itself.
To get a proper display of the archived print such external files must be accessible also when the print is recalled. However to enable moving of the print archive files between different machines, if any file is not present, TRIO will search in the following paths also:

- **The database path as given in the preferences**
- **The TMP path**
- **The current path given when the program was started**
- **The report definition path**
- **The path of the TRIO programs itself**

If not found anywhere, the display will just be left out without warning.

3.3.5. The SSV Printindex file

Also a printindex SSV textfile will be builded with one line for each printsession and the fields:

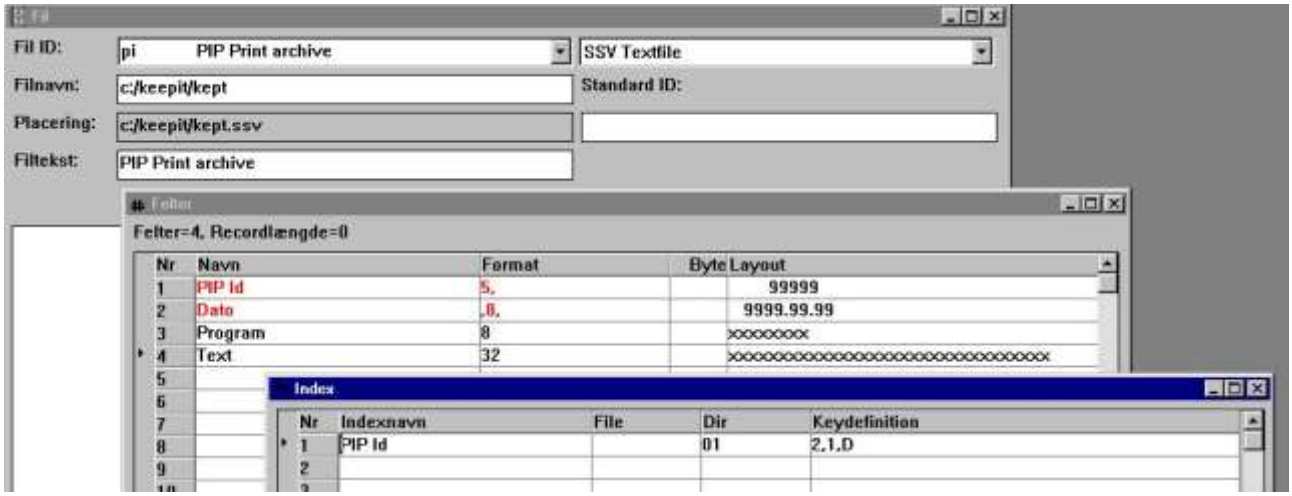
- Printsession number within the zipfile
- Date printed
- Report number
- Report name
- Additional user-controlled informations given in calculations
(as customer number, name, invoice number, date)

As an example the ssv file could contain:

SW-Tools
1;19990715;dm1001;Article report
2;19990715;dm1008;Invoice

3.3.5.1. Defining the printindex for individual print recall

If you want to use such a printindex file for searching and display with IQ you must first make a file definition like:



15. A standard definition for a PIP file

Note the index definition of 2,1,D causing the print to come in date order, the newest print first.

3.3.5.2. Calculations in IQ for individual print recall

Then define an IQ program on this file as pi#1-99L on listform and add the following calculations:

16. Calculations for PIP print recall

The SCRPRT function gets the first parameter as PIP id

c:/keepit/kept9908,1

and the second parameter controlling the display (see functions later) as:

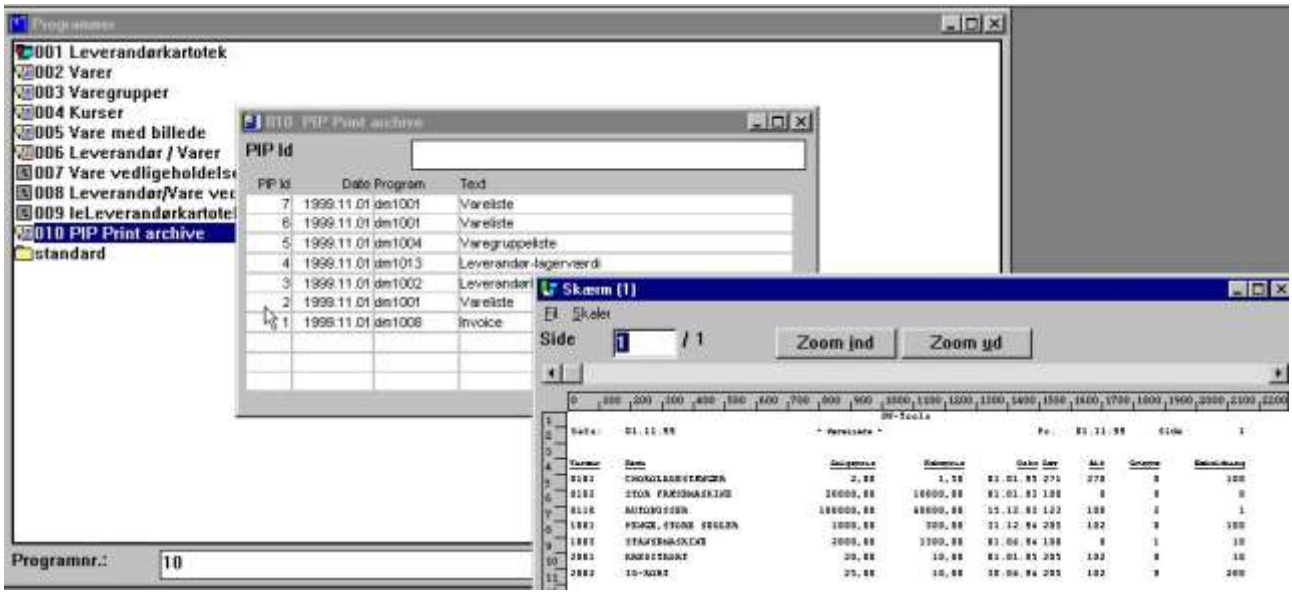
1 = Start display on page 1

1 = Display in window

-1 = Zoom factor, zoom out once to reduce size

which gives the result:

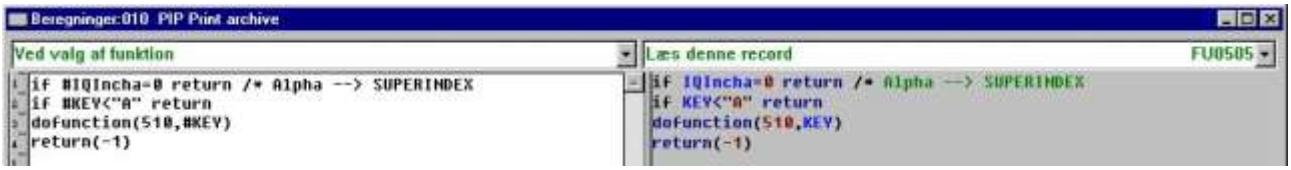
3.3.5.3. The IQ program for PIP print recall



17. The IQ program for PIP print recall

3.3.5.4. Calculations in IQ to ease searching

You may add the following lines to the IQ program



18. Calculations in IQ program for PIP print recall

by which IQ will switch to SUPERINDEX automatically when a letter is entered as searchkey, whereas a number input goes as normal to index search. Such routine may be used in general in IQ programs.

3.3.6. PIP Print archiving controlled from within the calculations

The calculation function PIP for print archiving has been introduced as:

PIP (fields, filename [,optional pages])

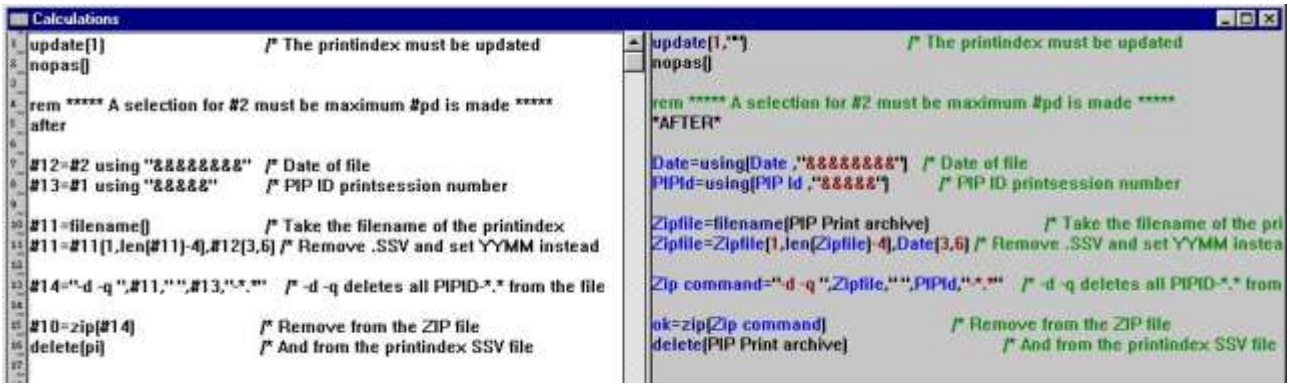
You may use this function in a report to switch on the print archiving for a report instead of for a printer. The PIP filename does not have to be set on the printer, if it is set anyway the calculation specification will overrule this. As an example:

LAST
PIP("#1-3", "c:/keepit/kept")

will archive the print and place the contents of field 1,2 and 3 in the printindex file (kept.ssv).
The first field in the printindex file will always be the PIPID or printsession number as to retrieve the print. If the filename given is 4 characters or less so that YYMM is added, the date YYYYMMDD is inserted as the next field. After this follows the fields you state for the PIP function or, if nothing stated, the report number and name.
The PIP function may be used in the NORMAL section of a report also to archive all pages printed since the last call of PIP. This means you may call PIP after finishing printing an invoice to get the invoices separated in the print index for searching by the print recall.
The last PAGES parameter of the PIP function may be used for further control of what to archive, could be totals only from a huge report, but should normally be left blank. See the PIP function description for details on this.
If a report results in more separate entries in the print archive, like one for each invoice, the last one of these will be displayed by clicking on the endtime in the start log.

3.3.7. Cleanup of old print

In our example above the zipfiles was named keptYYMM.zip, if you need space quickly it is just to delete the unnecessary old files from the disk.
However the correct method is of course to make a report on the printindex file deleting all old records and in the same time removing these from the print archive ZIP file. For this we need the ZIP function which we pass the zip filename and PIPID, together with the mode -d for delete and -q for quiet operation to ignore all error messages:



19. Calculations for a PIP cleanup report

The freefields 11,12,13 and 14 used above are all textfields of 128 chars. The field #10 will be 0 if the zip entry was deleted or 1 it not deleted.

3.3.8. PIP Locking and error handling

During the print archiving the printindex SSV file is of course locked, if more users try to archive at the same time to the same file the last one will get a warning and will have to wait until the first is completed.
To speed up the archiving, control is not handled back to Windows during the ZIP packing which may cause a short pause during the archiving.
The ZIP function itself may give error messages as disk full or report problems with the zip file.

4. Report layout definition and tuning

4.1. The layout function

The layout function has been modified on several levels. Most importantly, the rulers and alignments now support use of centimetres, inches and points.

4.1.1. New layout when report is defined

In previous versions it was almost impossible to select the same starting position of a field on the next line as on the line previously defined. This is now solved.
The box size calculated by the new report layout will now match the box size calculated later on in the layout function.
Please refer to the new preference for further details.

4.1.2. Autosize of fields

All TRIO standard reports will as standard insert fields with the option 'Autosize box' width and height (both included).

4.1.3. Alignment

When inserting or moving fields in the layout it is automatically aligned horizontally. If this is not required the checkmark of the new option in menu

File, Edit, Align to column

may be removed.

4.1.4. Auto scroll when moving, inserting items in layout

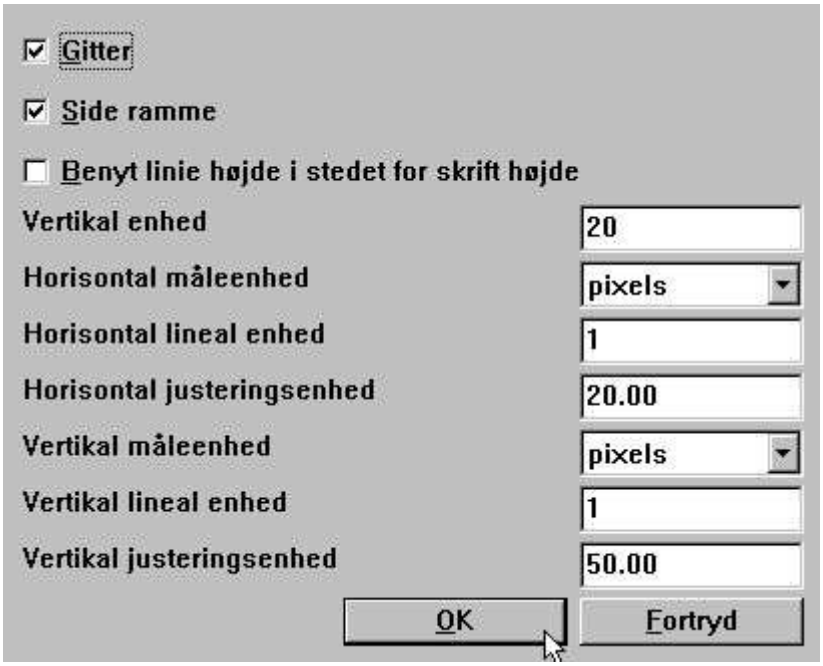
The layout function will automatically scroll the layout when the mouse cursor reaches the edges of the layout window while moving or inserting items.

4.2. Layout preferences

The layout function contains a preference dialog. It is activated through the menu

File, Preferences...

and allows you to setup how you need to work with the layout. The changed settings will be saved individually on each report.



20. The preferences in the layout function

4.2.1. Display options

4.2.1.1. Grid

The layout grid is provided as guideline. You may turn this option on and off. How the lines are displayed depends on the horizontal and vertical ruler step.

4.2.1.2. Page frame

The page frame is provided as guideline. It will display the size of the page dependent on the printer currently selected in the layout. It will NOT limit you from exceeding the page when inserting and moving items.

4.2.1.3. Use line height instead of font height

This option is only used in connection with a report. Normally, inserting a field uses the standard font height to set the box size of the field. However, if you want to use the normal line height of a report you may turn this option on.

4.2.2. Vertical position step

The vertical step is only used in connection with the layout of a new report. When you create a new report and do a click on the horizontal measurement bar you change the starting position of the next field to insert. This position will be aligned according to the vertical position step. Doing so makes it easier to get the same starting position of fields when creating reports with more than one print line.

4.2.3. Horizontal/vertical measurement, ruler and alignment

The measurement may selected as

- **pixels**
- **cm**
- **in**
- **points**

The preference dialog with automatically convert the existing alignment value according to the selected measurement.
The alignment value is used when inserting and moving items within the layout. An item will be aligned according to the value setup here.
BE AWARE that changing the vertical ruler and alignment step to anything but pixels/50 may cause you to insert or move items to positions that DOES NOT correspond to the correct print line used be the report generator.

4.3. Additional picture formats

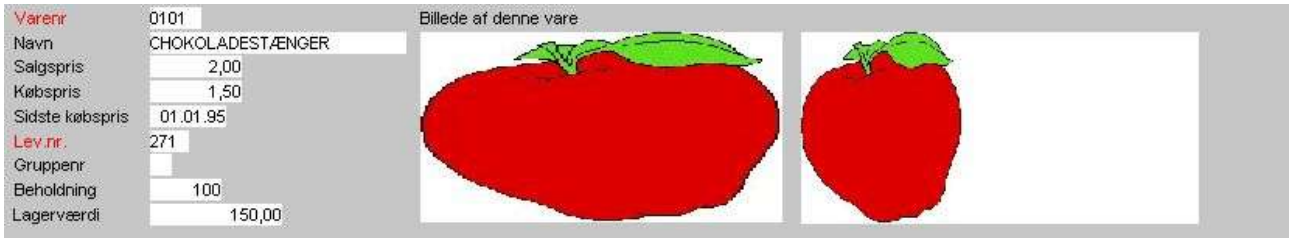
TRIO supports display and print of the following picture formats

BMP - Microsoft Windows bitmaps
CUT - Dr. Halo
DIB - Microsoft Windows Device Independent Bitmaps
GIF - Compuserve Graphics Interchange Format
IFF - Interchange File Format (Amiga Electronic Arts Deluxe Paint)
IMG - GEM Raster files (Digital Research)
JPG/JPE/JPEG - Joint Photograpics Experts Group
LBM - Interchange File Format (Amiga Electronic Arts Deluxe Paint)
MAC - Macintosh Paint
MSP - Microsoft Paint
PCX - ZSoft PC Paintbrush
PIC - Pictor PC Paint
RAS - Sun Raster files
TGA - Targa TrueVision Files
TIF - Tagged Image File Format
WMF - Windows Meta-files

However as there may be different variants of these picture formats, it is just impossible to test and support all formats in all variants, especially with respect to different types of compression. In case you experience problems with any type it may be necessary to convert into one of the other types or save with different compression level.
You should note that display of pictures might be of essential better quality and faster in the 32 bit version than the 16 bit version can do.

4.3.1. Scaling of pictures

Pictures may be displayed either as
a. Filling out the box given in the layout, which means the picture is stretched in both directions or
b. Sizing inside the box as large as possible, but keeping aspect ratio of the image, which may cause either left side or bottom of the box to be left blank.



21. Picture displayed with or without maintaining the aspect ratio

For the time being, the WMF is always displayed according to rule a, all other according to rule a or b.

22. Keeping aspect ratio

4.3.2. Animation of pictures

You may create an animated picture in IQ just by displaying the same picture field multiple times with different contents, note this requires a rather fast PC.
The DISP function to redisplay one or more fields as DISP(#30) ensures that a field is redisplayed after calculations is done, before any new input is made. For picture animation you will have to add the U option as DISP(#30U) to get the screen updated straight away without waiting until next input.

4.4. The database window, display and search

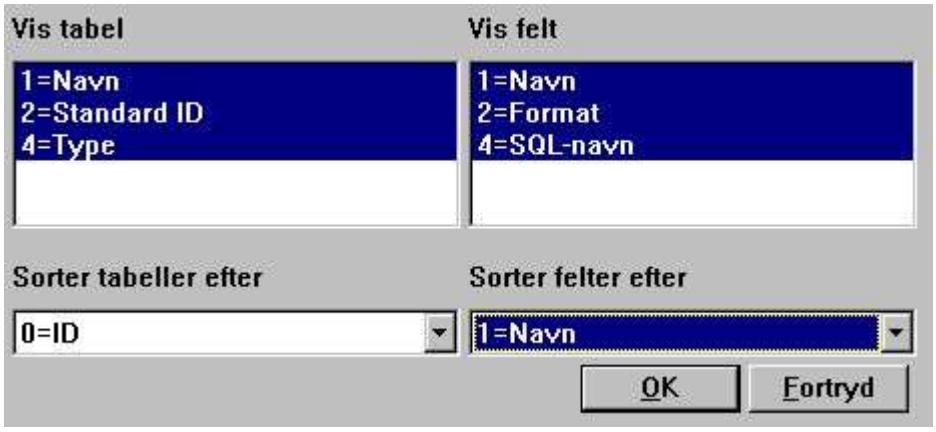
The database window include settings for which information to display and how to sort the information within the window. Furthermore, it is possible to search the entire data dictionary for fields or tables.

4.4.1. The display and sort of the database window

The options you may setup for the database window is found in the menu

View - Options...

when the database window is active.



23. View and sort options of the database window

Apart from always including the table id, the view/and sort of tables include

- **Name**
- **Text**
- **Standard ID**

Field always includes field numbers, but may also view and sort

- **Name**
- **Format**
- **SQL name**

If the option of viewing fields includes field format the database window will display the following:

24. Database window including field format

4.4.2. Searching the database

When searching the database definitions, it is possible to find tables and/or fields. The search option is selected when the database window is active in the menu

Edit - Search

25. Entering the search criteria

The search is performed according to the search method selected, which may be

- As entered
- Case-sensitive
- MATCHES/As entered
- MATCHES/Case-sensitive

The MATCHES allows you to work with part of text following a specific syntax. The syntax is based on the use of some special characters

*** = any text**
? = any one character

For example, the database definitions may have several tables, where fields are defined as dates. However, only searching after the text

date

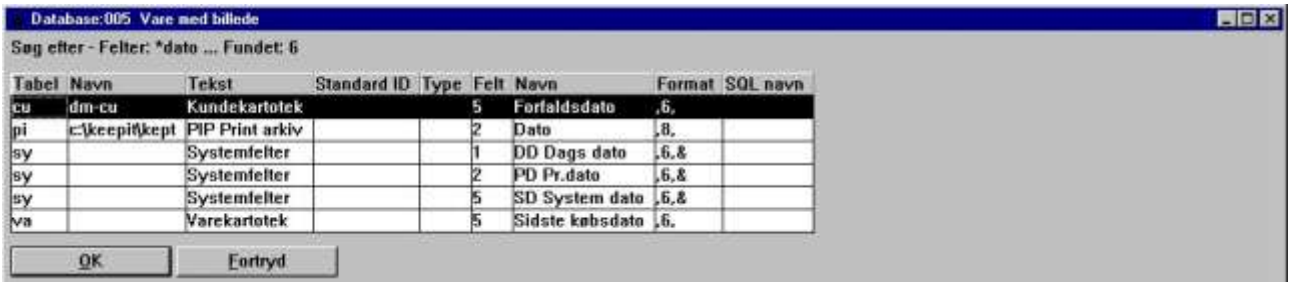
will not find a field in upper case if the method is as entered. Also, it will not find a field

date of last purchase

If using the method MATCHES/As entered and searching for

date

it will find any field having date within the field name, e.g. any text followed by the text date and ending with any text.



26. The search result shown in the database window

4.4.3. Display of used files only

This option is activated from the file menu when the database window is active.
You may select to show used files only for the program in order to get the overview of needed files as short as possible.

4.4.4. Actions by click on a field in the database window

When defining selections or calculations you may click on any field in the database window to get the full field definition shown. By selections the field will be inserted in the current selected box on screen also.

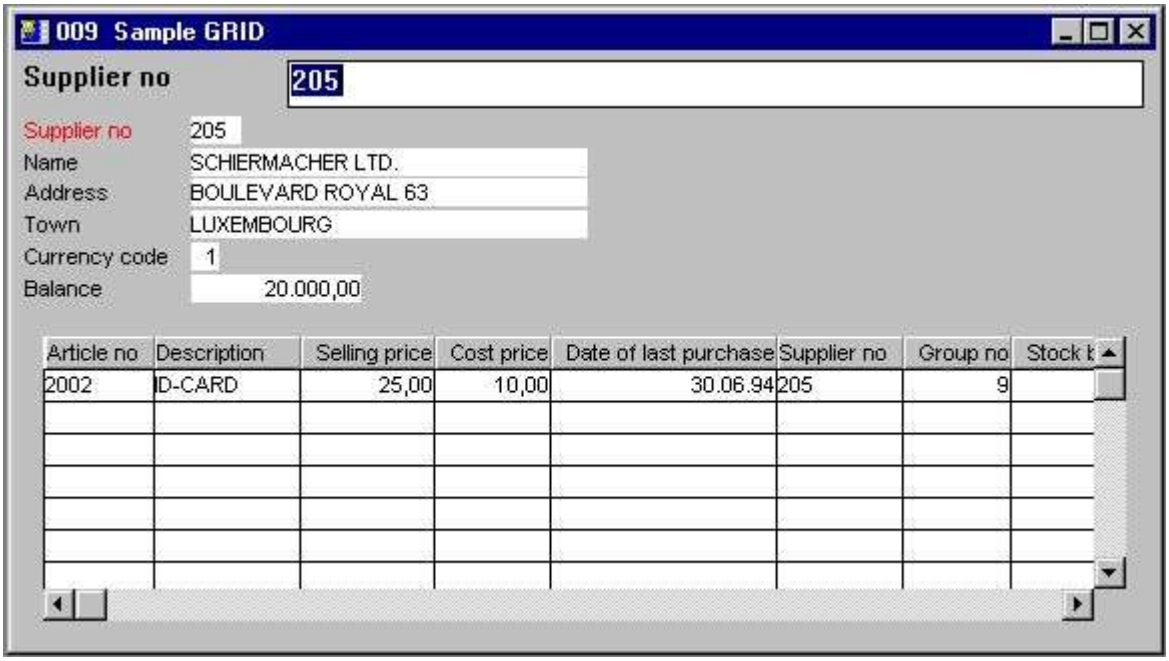
4.4.5. The function parameter wizard help

By using the function parameter wizard by calculations (ctrl.Q), you may go directly to the windows manual section for this function by selecting HELP (F1).

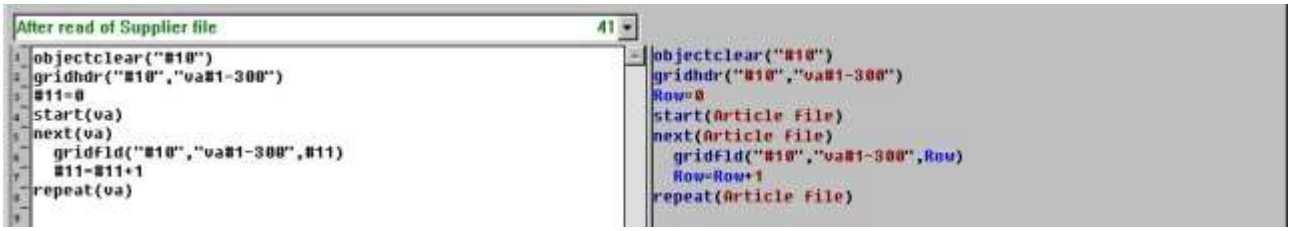
5. IQ

5.1. GRID listboxes

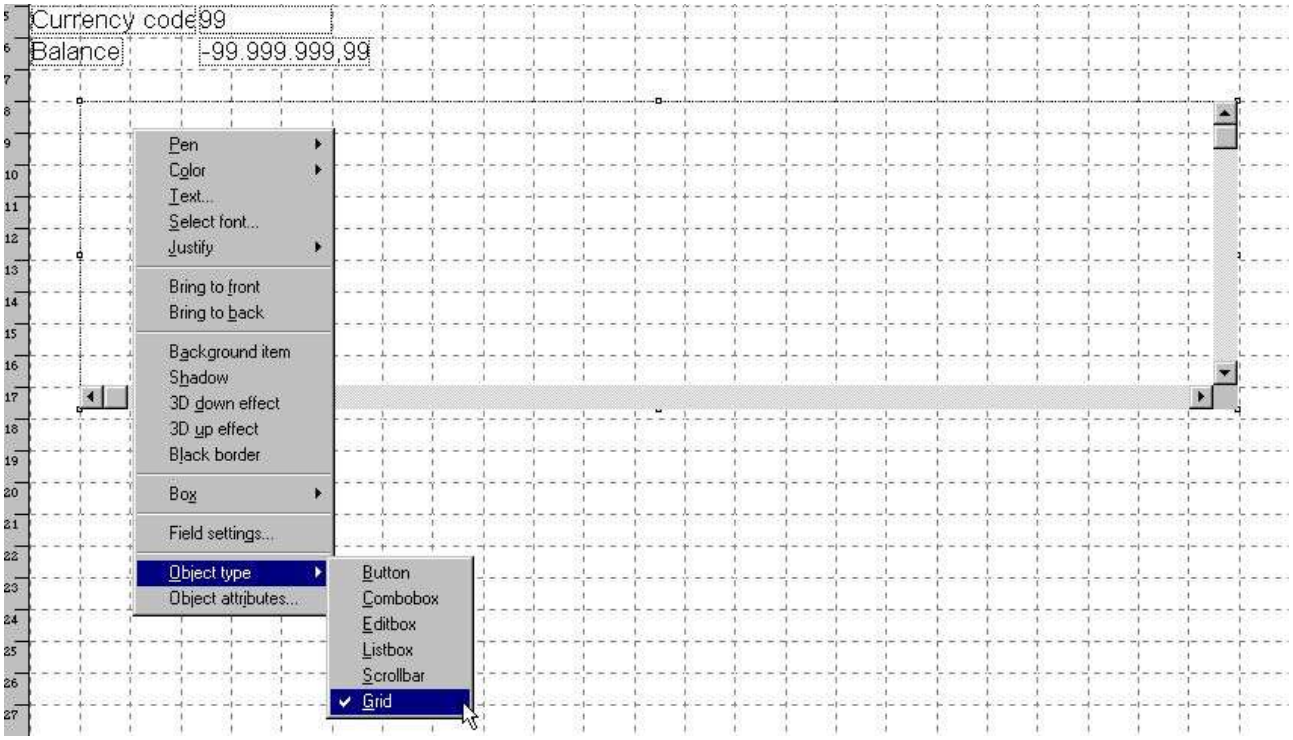
The GRIDHDR and GRIDFLD function has been added.



27. Sample use of GRID



28. The GRID calculations



29. The GRID field in the layout

5.2. DGRID for simple data selection

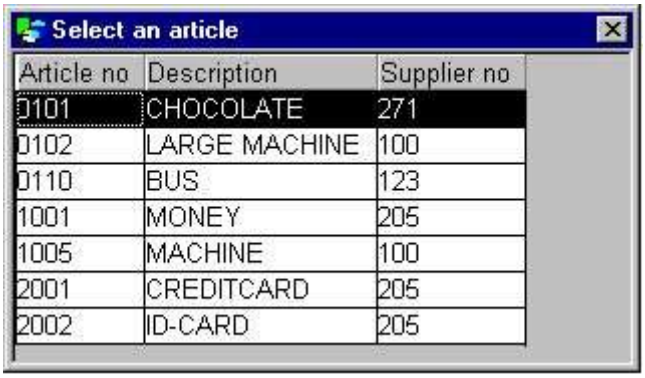
The subfunction DGRID may be used to select a record from a dialog box which will block the current application until selection has been made.
For example, by entering the following calculation by click on a field in a IQ query

DGRID("va!1-2,6!1!0!Select an article",#50)

will read records from the article file 'va' and display the fields '1-3' which is sorted according to the index number '1'.
The function will return the value of the displayed field '0', e.g. field number 1 because

Displayed field	Field
0	1
1	2
2	6

which is stored in field #50.
Finally the dialog title is set to 'Select an article'.

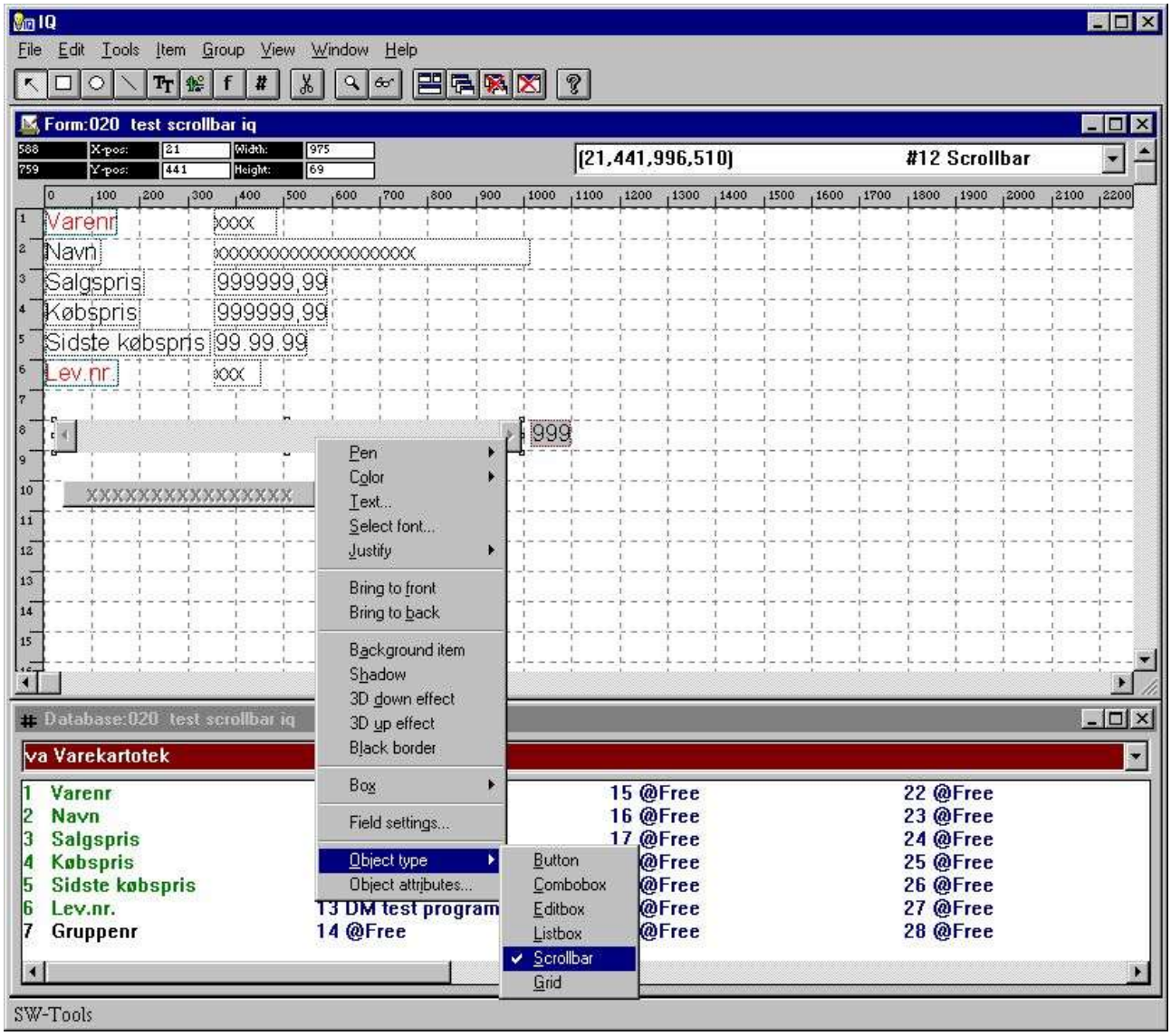


30. DGRID dialog

For more information please refer to the DGRID subfunction.

5.3. SCROLLBAR object

The scrollbar object may be used to display numeric values from 0 to 32767.
To make a scrollbar field you may define a work field named 'Scrollbar' with format '6,' and change the object type in the IQ form to 'Scrollbar' when inserted.



31. Scrollbar inserted in a IQ/DM form

When using a scrollbar in IQ the following calculations are required:

- Setting the maximum value using ObjectAddString()
- Getting the current position of the scrollbar using ObjectGetString()

As an example you may define the work field #12 as 'Scrollbar' with format '3,' and add the following calculation by start of program

ObjectAddString("#12","100","")

which will set the maximum position of the scrollbar to 100. Then a calculation by click on field #12

#12=ObjectGetString("#12")

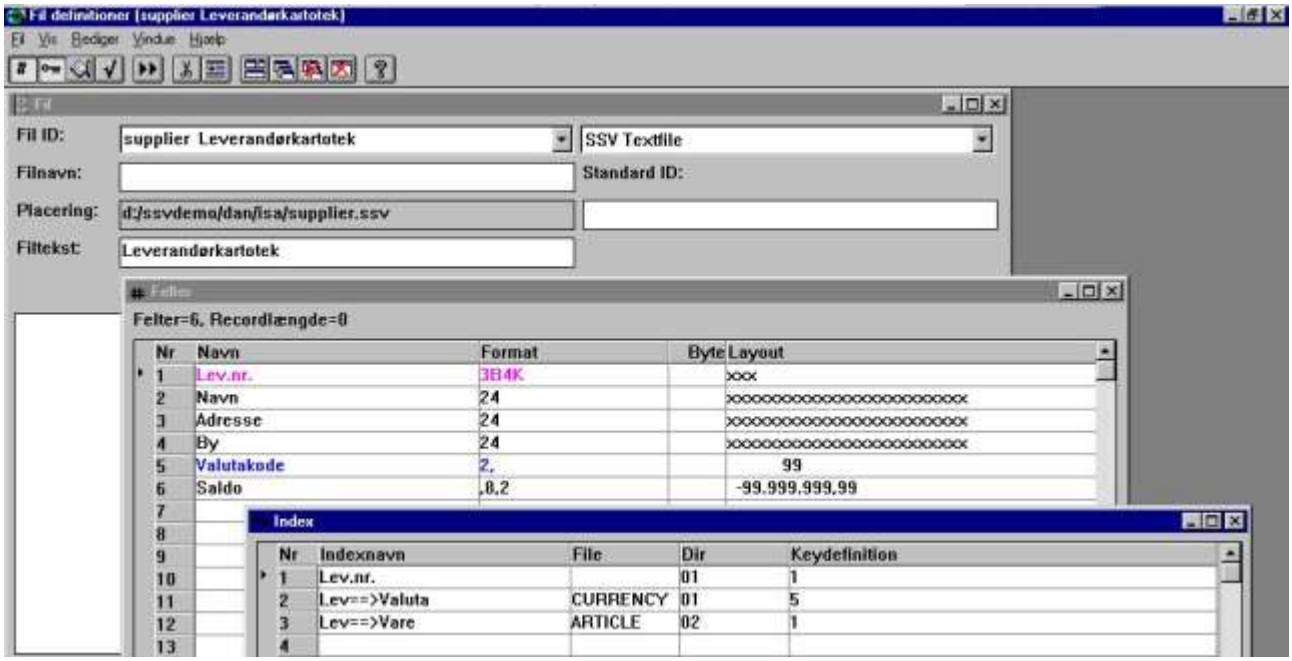
in order to retrieve the current scroll position.

When using the scrollbar object in DATAMASTER you do not require to use the ObjectGetString() to retrieve the current position, because DATAMASTER does this automatically by setting the value into field #12.

6. The datadictionary

6.1. Long datadictionary file ID

The 2 character file ID has been extended to a maximum of 8 characters.



32. The demo supplier file with long file ID

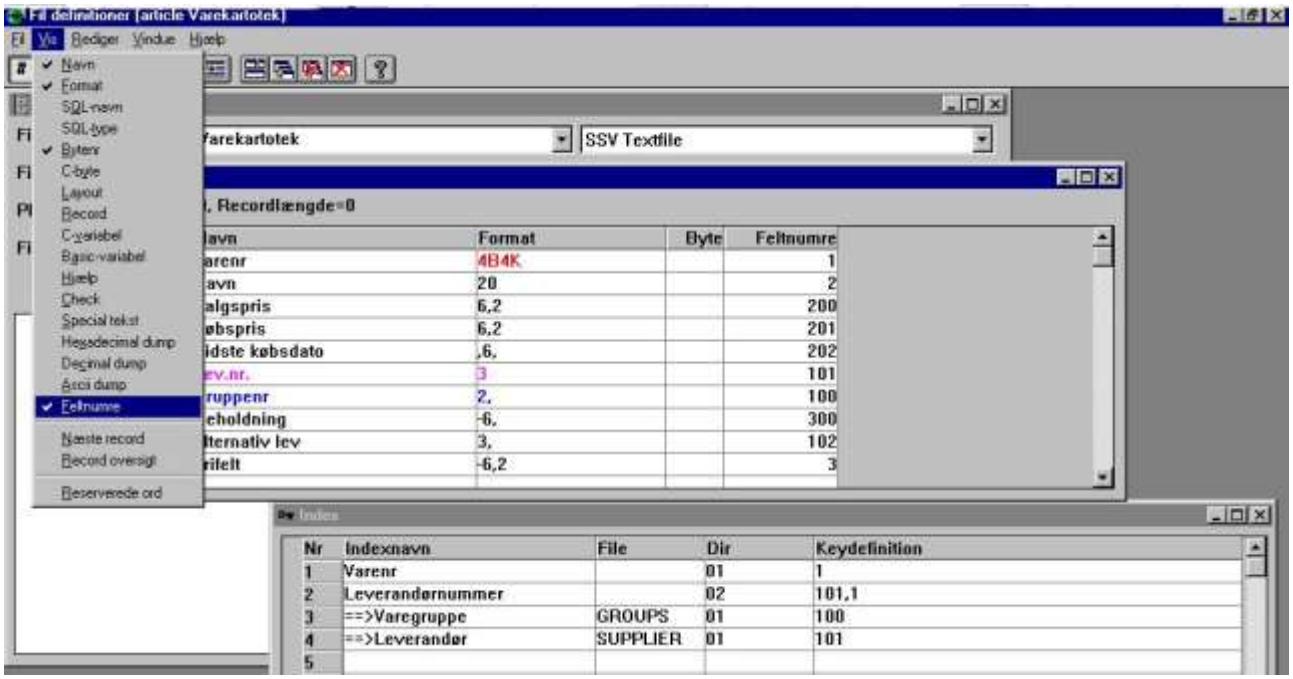
Note that connections, READ(....) and all file/field access in general is extended in this way, and use of fileIDs longer than 2 characters is in no way backward compatible with previous versions, also with respect of any Unix server.
In calculations you may then give fields as:

va#47=1
customer#47=1
invoice#47=1
invoice#statcode=1

The use of upper/lower case letters for the same file multiple times in a single program is limited to the 2 first characters only, (auftrag/AUFTRAG/Auftrag/aUFTRAG), more than 4 times the same file is not considered relevant.
On 32 bit systems you may even create fileIDs with up to 32 characters, but this is not recommended as all compatibility with the 16 bit version will be lost.
A conflict with old TRIO programs and long filenames exists in the syntax in read of files, where the connection is stated explicitly, as READ(va.le), the old syntax READ(vale) without point is still supported unless a long fileID 'vale' really is present.

6.2. Datadictionary fieldnumbers

Each fieldnumber in a file may be stated separately independent of the real sequence, as a new column in the FDF has been added for user-defined fieldnumbers:



33. Amending the fieldnumbers in the datadictionary

In the database window the fields are shown sorted by the new sequence and all references go to the new fieldnumber, for example should the db now be calculated as #310=#201-#200

34. The database window after fieldnumbers has been amended

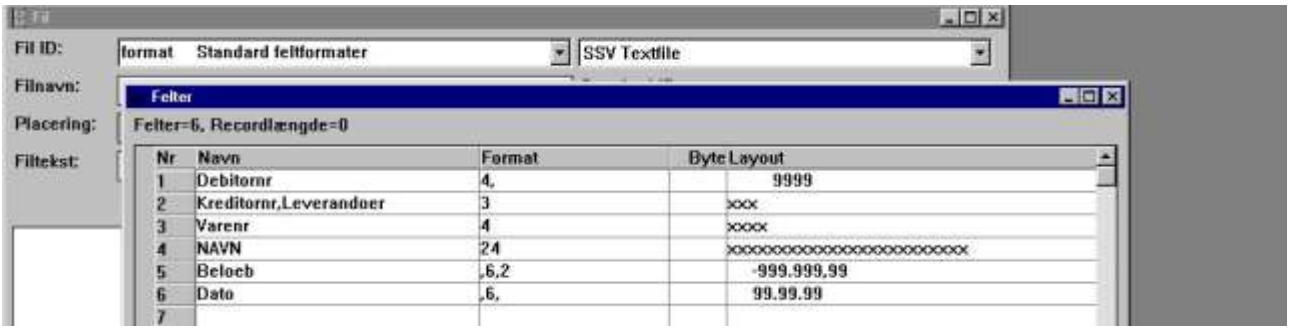
Note that workfields are numbered as usual after the last used field in the file.
The maximum number of fields in one file is still 999, but fieldnumbers may be up to 2.000.000.000 in the 32 bit version. The 16 bit version is restricted to maximum 65000.

6.3. Datadictionary standard field formats

A field format may be stated as AMOUNT, the real format will be taken from the file FORMAT if present by searching for a fieldname Amount.

6.3.1. The standard FORMAT file

To activate the standard field formats, just create a file named FORMAT in your datadictionary like:



35. Creating the FORMAT file with standard formats

The fieldnames given here will be the standard formats, these must be alphanumeric without special characters, upper/lower case does not matter. You may state synonyms by separating these by comma as shown for supplier.
The field formats may contain bytenumbers, packtypes and other options just like a normal field format.

6.3.2. Defining a file using the standard formats

You may use these standard formats when defining a file like:

Nr	Navn	Format	Byte Layout
1	Varenr	VARENR B4K	xxxx
2	Navn	NAVN	xxxxxxxxxxxxxxxxxxxxxxxxx
3	Salgspris	BELOEB	-999.999,99
4	Kobspris	BELOEB	-999.999,99
5	Sidste købsdato	DATO	99.99.99
6	Lev.nr.	LEVERANDOER	xxx
7	Gruppenr	2,	99
8	Beholdning	-6,	-999999
9	Alternativ lev	LEVERANDOER	xxx
10	Frifelt	-6,2	-999999,99
11			

36. The demo article file using standard formats

The standard format is used when a field format starts with an alphabetic character. The standard format name goes up to the first space met, after this you may state additional options for the field as shown for the first field with the K=Key and B4=Bytenumber 4 options added.

6.4. Integration between different database systems

As it is being more common to have different database systems on the same equipment, it is possible to define these completely separate but still access and mix these from within the calculations.
A qualifier stating the ID of the database followed by underscore may be stated in front of any fileID as:

comet_invoice = Filedefinitions COMET, file INVOICE
alx_customer = Filedefinitions ALX, file CUSTOMER

Which may be used in calculations as:

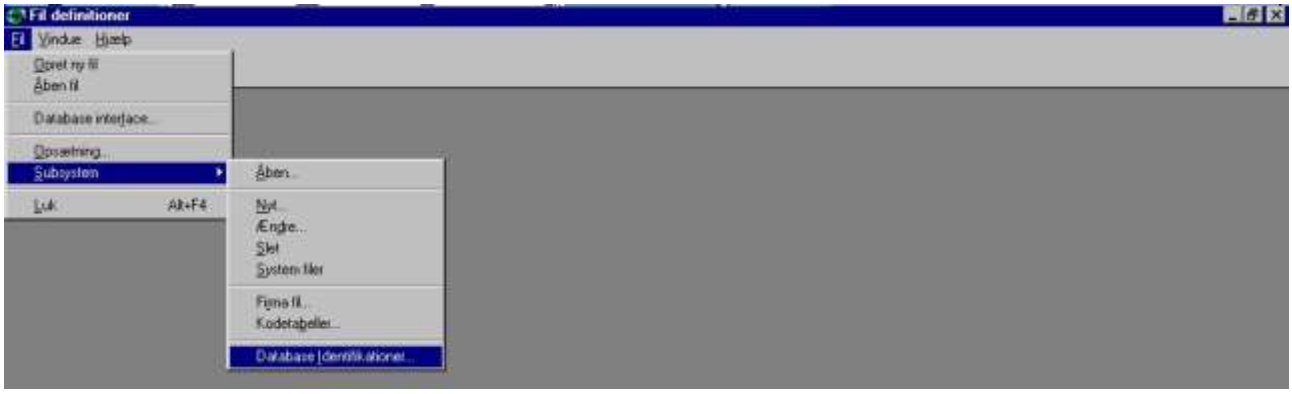
```
alx_customer#47=1
READ(comet_invoice),alx_customer#47
comet_invoice#statcode=1 .
```

6.4.1. The database system IDs

A file containing the database systems ID (BASID.SSV) states the connection from a fileID like 'comet_xxxxx' to a directory with the file definitions (FILES.SSV/xxxxx.SSD and BASIS.SSV) for the database 'comet'.
The BASID.SSV file must always be placed on the same directory, as where the BASIS.SSV file with the drivers are found for this database.

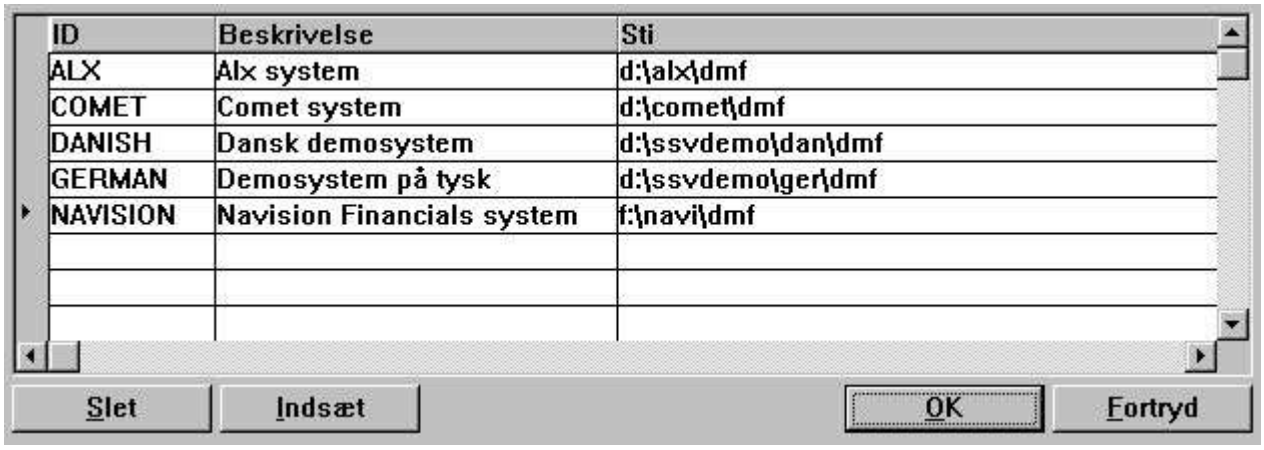
6.4.1.1. Creation of database identifications

You reach the database ID's from the FDF subsystem menu:



37. Selection of maintenance of the database ID's

wherefrom you get the following screen:

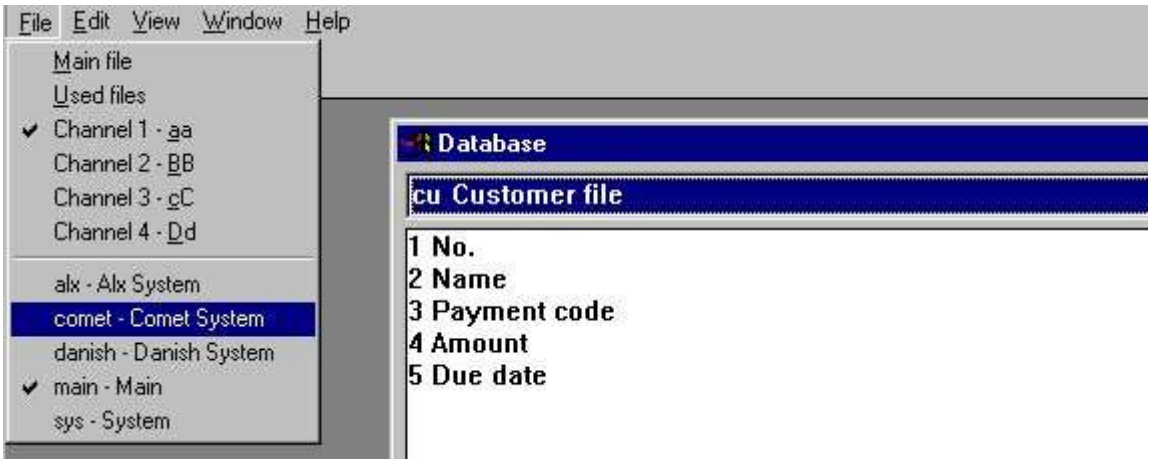


38. Maintenance of the database ID's

Note that the database ID is alphabetic and uppercase (A-Z) only, you cannot use digits or special characters.

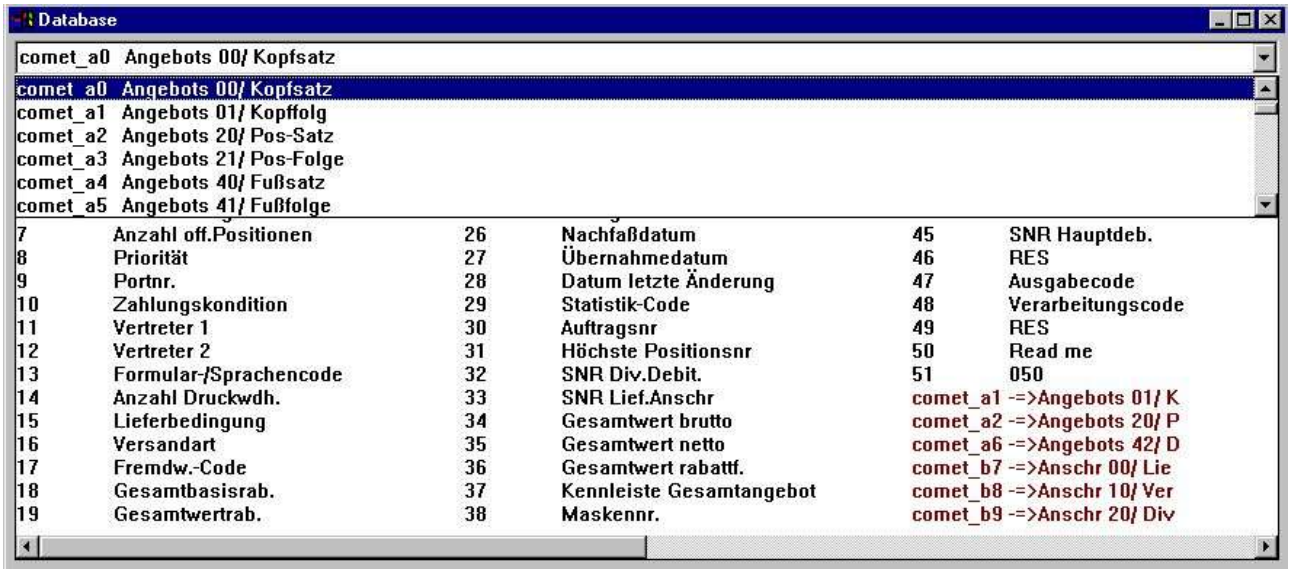
6.4.1.2. The database window when BASID is activated

When more database ID's is present in the system these will show as menu items in the file menu for the database window:



39. The database window file and database selection

When you select for example the COMET database, you will get this as:



40. The file, field and connection selection in the database window

6.4.1.3. The fixed database ID's MAIN and SYS

TRIO will always add two ID's:

- **MAIN**
- **SYS**

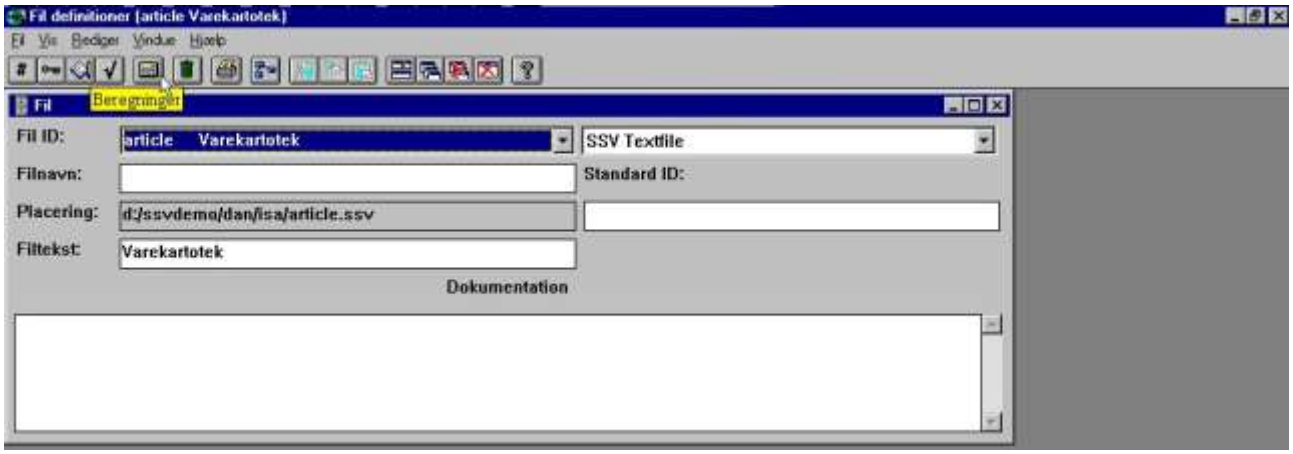
MAIN you may use to return from a selected database to the original.
SYS is more for programmer's use, where special system fields may be added. This is also used internally by TRIO itself.

6.4.2. File connections spanning over different databases

In the FDF standard file connections may be defined to other database systems, as seen above in the database window the connected file is just stated as BASID_FILEID to get another database.

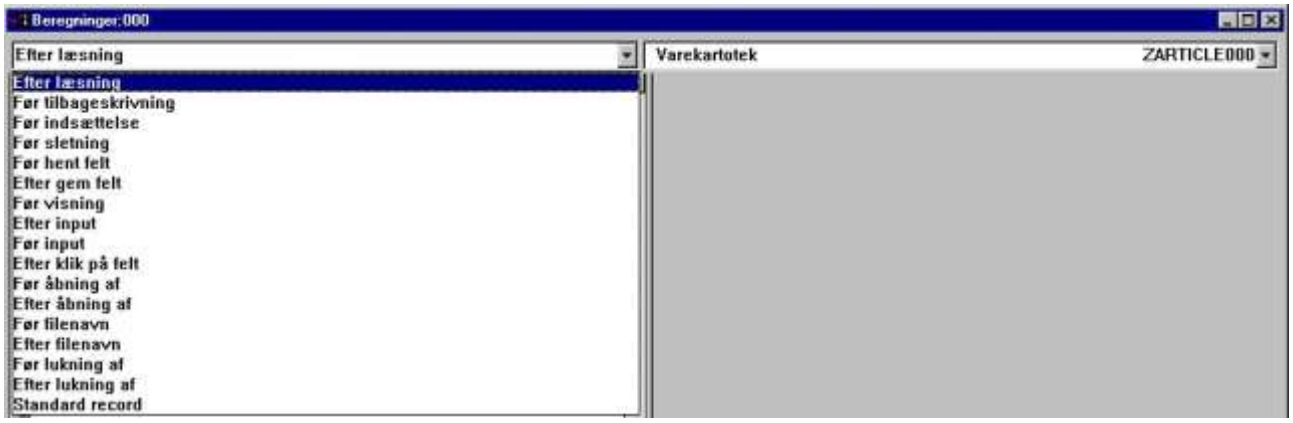
6.5. Datadictionary calculations / Stored procedures

Each file or field definition may be associated with one or more calculation blocks held in the datadictionary itself. These stored procedures are defined by:



41. The new calculation icon in the data dictionary

Like the present IQ calculations the calculation block name decides, when the specific calculations should be executed, e.g. by READ or by WRITE of the file.



42. Selecting where to calculate for a file definition

When the calculation block is selected, you may select a single field also:

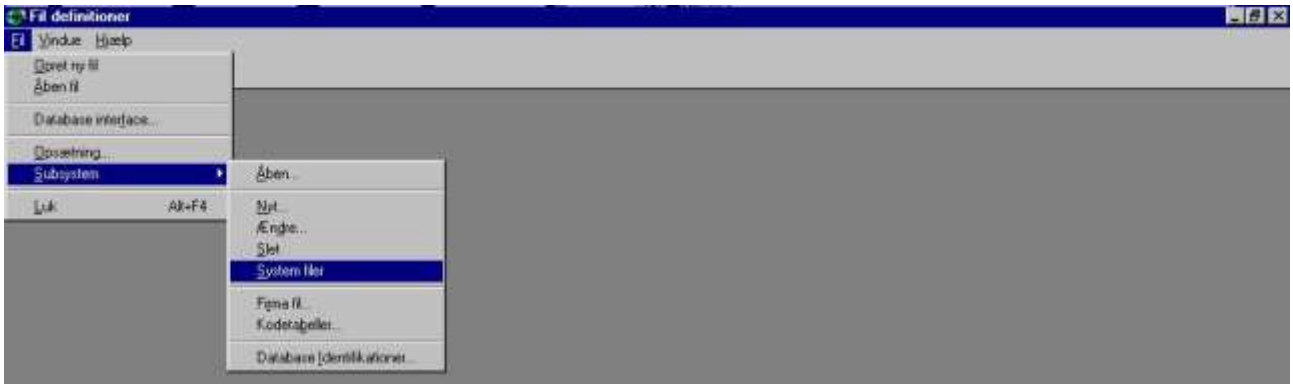
43. Selecting calculations for a single field or the complete file

6.6. User definable subfunctions

Common blocks of calculations may be defined as a subfunction like the function descriptions which comes with TRIO in the files 00-99.
These files are made like normal file definitions, but may hold function definitions and documentation of internal TRIO functions and external DLL functions which may be made by other programmers.
From version 008 you may now also define your own set of functions as common calculations, which may be modified and parameterised when loaded.

6.6.1. Selecting the function definitions

You reach the function definition files when the FDF program is switched into the SYSTEM FILES mode:

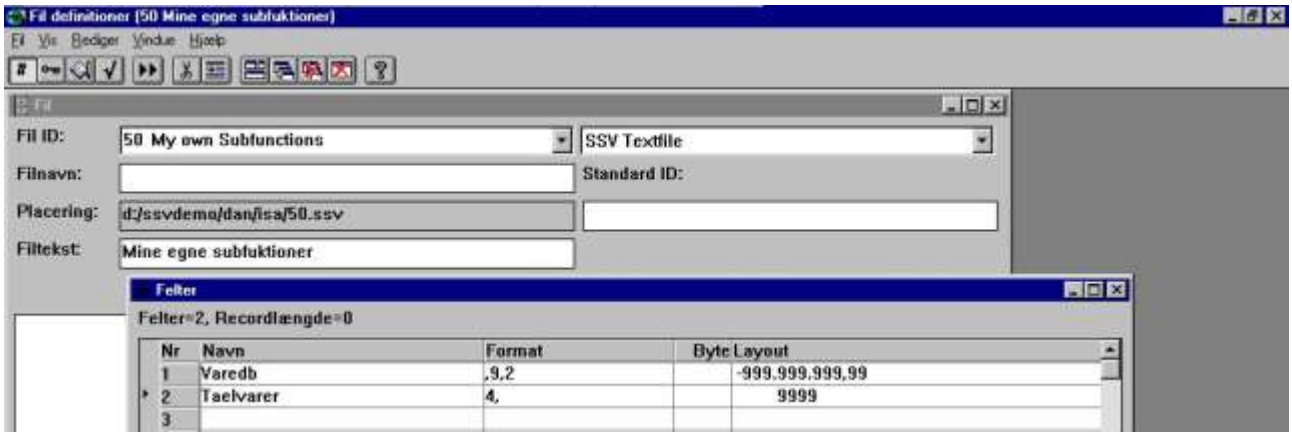


44. Selecting the SYSTEM FILES mode in the FDF

which enables you to maintain / create file definitions starting with a digit, which is normally blocked in the FDF.
You should change/create filedefinitions in the range 50-59 only to avoid conflict with this and future TRIO releases.

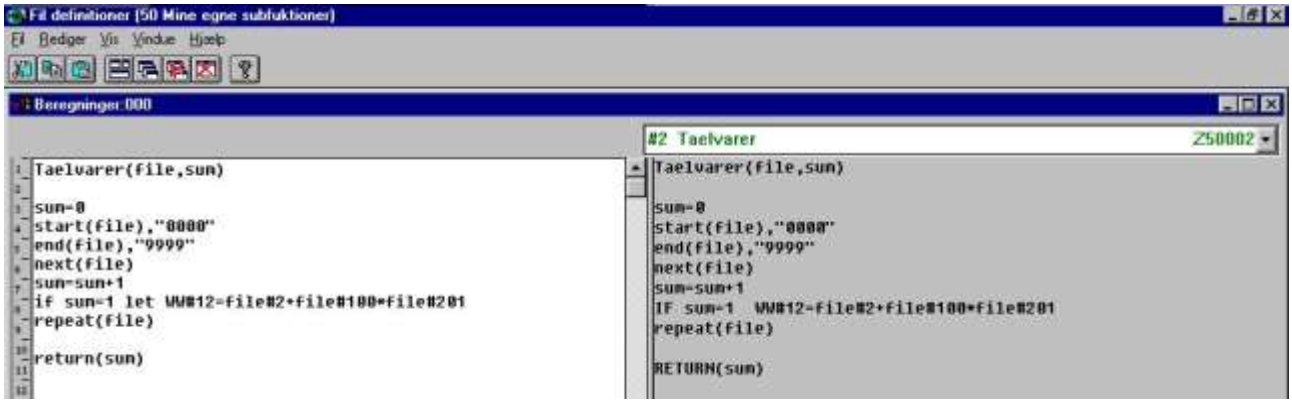
6.6.2. Creation of new functions

First the functions are created just as fields in the datadictionary, the fieldname becomes the function name and the field format states how the return value is given:



45. Creating new functions in the subfunction definition 50

The function- (field-) name must be plain alphabetic, no spaces or special characters is allowed. Upper/lowercase does not matter.
The calculations for the functions is then entered as the datadictionary field calculations:



46. Entering the function calculation lines

And the function is now installed and may be used in other datadictionary calculations or IQ as:

47. Using the homemade function

6.6.3. How it works

The first line of the function definition contains

functionname(par1,par2,...) .

When the function is to be calculated, the code is loaded into the program and any occurrence of par1 replaced by the first given parameter, par2 with the next and so on, whereafter the code is executed just as it was entered in the program itself.
In the above example the first lines becomes by load:

```
#10=0
start(article),"0000"
end(article)
:
:
```

7. External packages and runtime licences

A package consisting of a set of TRIO programs may be made and licensed individually by a certified TRIO distributor.

7.1. Packages

A package consist of one ZIP file named ddddllll.zip containing all necessary files, dddd being the distributor number, llll a package number.
The ZIP file may contain report definitions, IQ program definitions, file definitions, SSV files for parameters and pictures.

7.1.1. Package numbers

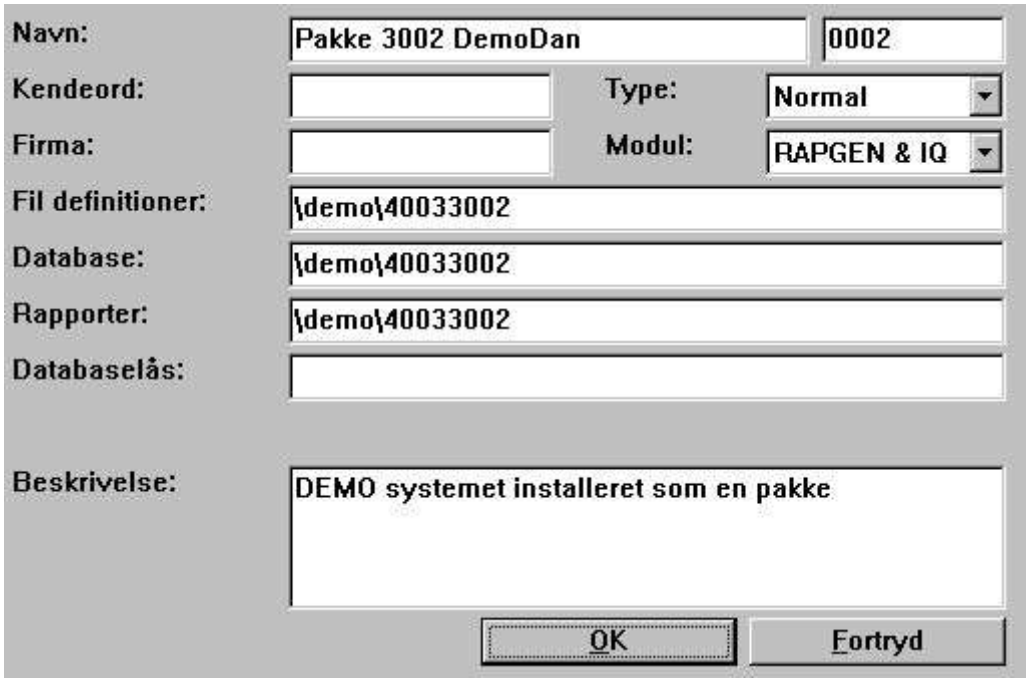
When the package number llll is in the range from 3000 to 3999, the ZIPfile will be crypted and can be used only by customers having a valid license for the product llll (3000-3999) issued by distributor dddd.
If llll is in the range 3000-3499 the license procedure for user ID registration is not necessary, 3500-3999 requires a final license to be issued for each PC user ID within 3 months just like the TRIO package.
If llll is any other number the ZIPfile is not crypted and usable by any TRIO customers with a full license, not just runtime, and the package does not have to be licensed separately.

7.2. TRIO Runtime licenses

TRIO may be delivered with a RUNTIME license, in which case it is not possible to create/amend programs at all, only start of premade programs within a licensed package is possible.
The FDF module will however be fully available also with just a runtime license, as the installation may require individual file setup.

7.3. Activating a package in TRIO

If a package 3002 is made by distributor 4003 the ZIP filename for this becomes 40033002.ZIP
This zipfile must be copied to the customer system, for example to the directory DEMO as \DEMO\40033002.ZIP
You may now set any report/IQ/FDF or database path in TRIO to point to this zipfile, either by a user homepath or creating a subsystem as shown:



48. Setting up a package subsystem

Above the file definitions is also to be taken from the package, you may however set these to be taken from the customer own definitions outside the package just by changing the path.

7.3.1. How TRIO handles a package

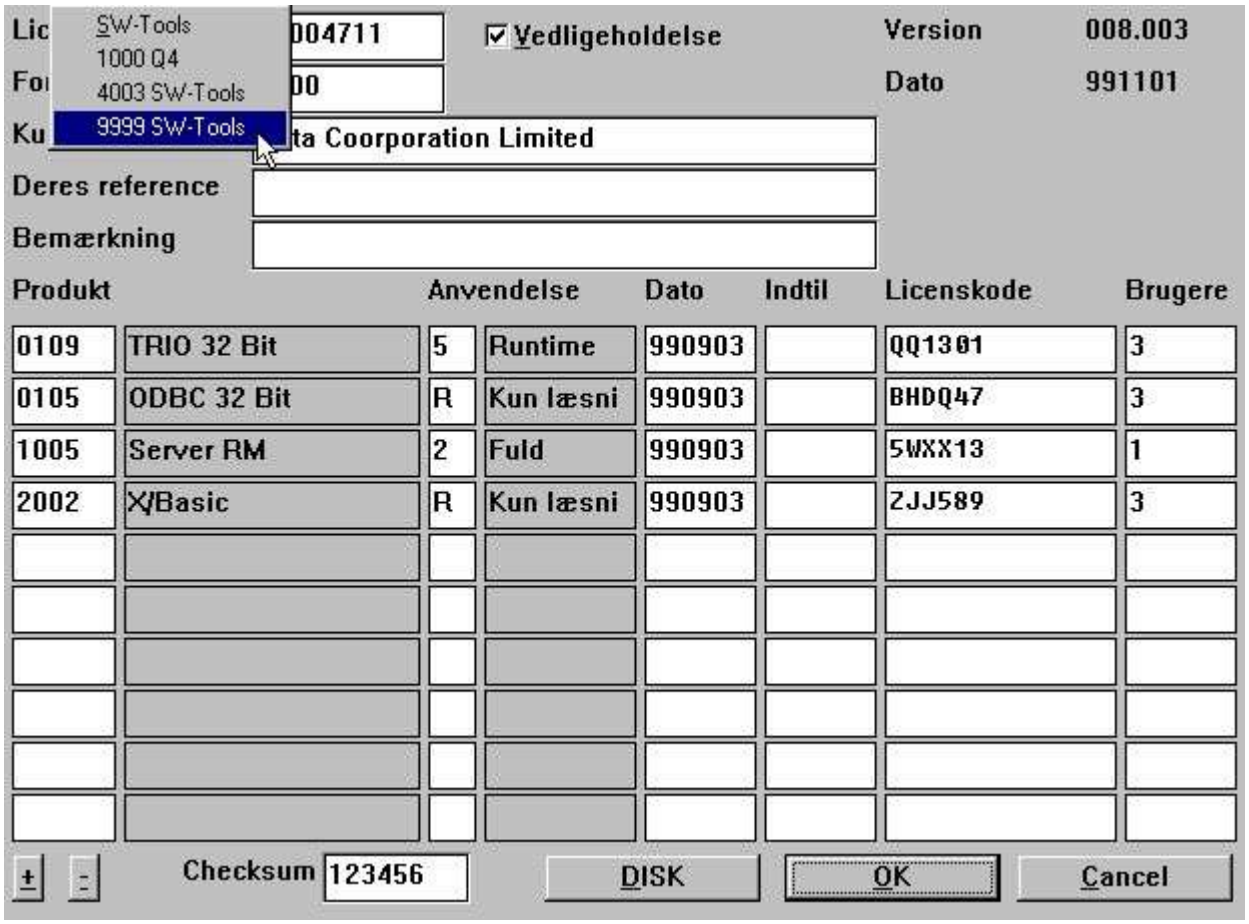
TRIO will for the above subsystem try to open a file for, say report number 777, like \DEMO\40033002\dm1777.src. If this file is not present and the directory stated is 8 digits, TRIO will check if the package file \DEMO\40033002.ZIP is present, if so the license is checked and the requested file is unzipped and encrypted from here without the user noticing this at all.
If the report is then amended and saved, the directory 40033002 will be created and the real file \DEMO\40033002\dm1777.src will be written. As the file is now present, all further access will go to the amended file and the original file in the package is ignored. The one program has now been moved out from the package and modified for individual use.
This means by later upgrading with a new version of the programs inside the package by replacing the ZIPfile, all individual amendments will be kept unless such files are removed. Also the package may contain a set of standard file definitions, if any file on the customer requires individual setup, it will just be moved out from the package ZIPfile when modified.

7.3.1.1. Package licenses

The distributor 4003 may issue a RUNTIME (or READONLY, which is the same) license for the package, in which case only start of the programs is possible, TRIO will never as described above allow amendments and move a program out of the package. File definitions however may be amended.
The distributor 4003 may also choose to issue a FULL license, in which case the programs may be modified freely if the customer also has got a full TRIO license.
Note that package numbers 3000-3499 do not require user registrations, maximum number is the same as for the TRIO license, 3500-3999 do, and all other numbers do not require a separate license at all and is always for full usage.

7.4. The license input program

A package license is considered completely separated from the TRIO license, and for this purpose the license program has been extended with a menu for selection of distributor:



49. Selection of package distributor in the license program

The first menupoint SW-Tools will be present always for the TRIO license, the additional distributors are shown if a distributor license file is present.

7.4.1. The distributors license file

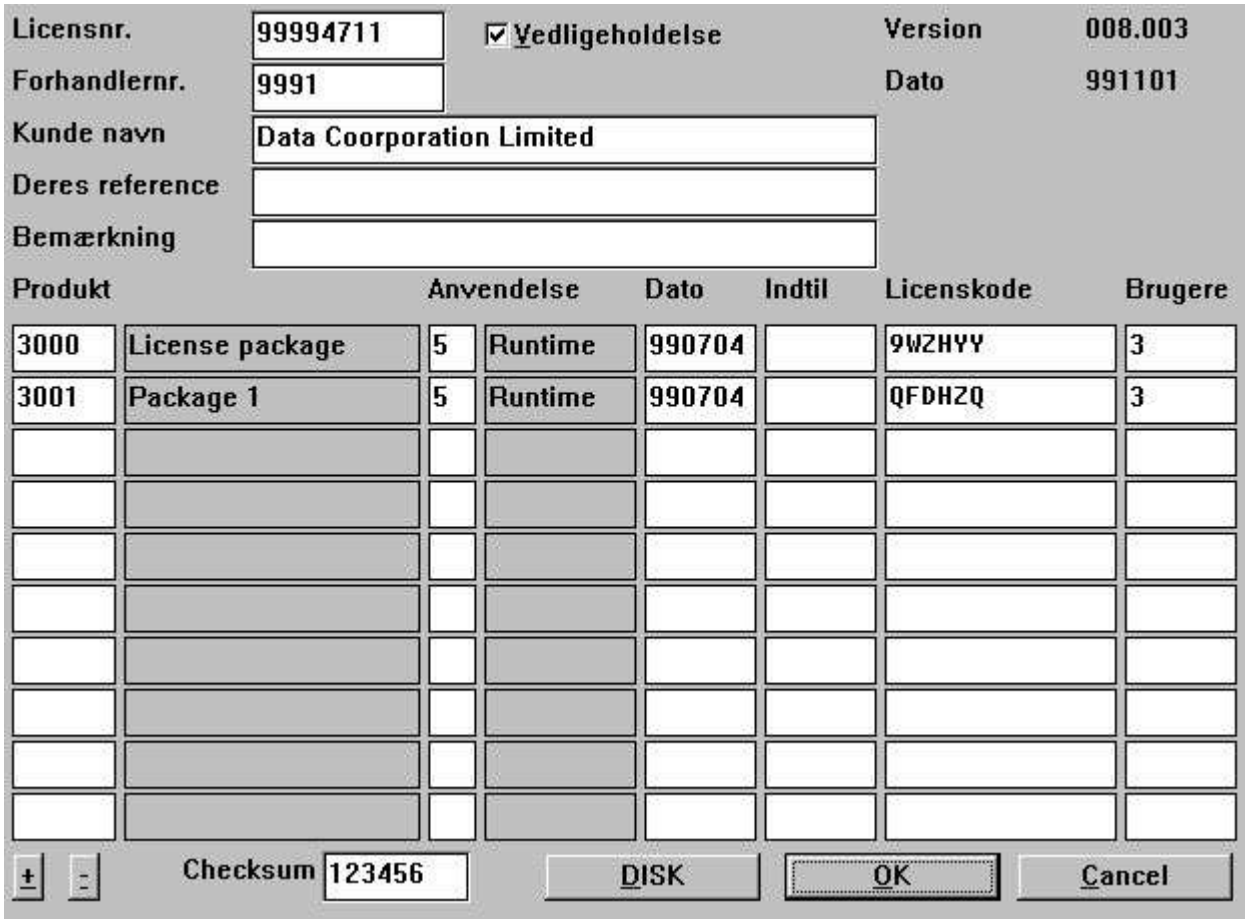
A distributor must identify each package with a package number and a name.
For TRIO itself the license program takes these informations from the file LICENSPR.ENG on the TRIO installation directory, ENG being the language suffix (ENG is also the default if the used language is not present)
The file LI4003PR.ENG must be made by the distributor 4003 and placed together with the LICENSPR.ENG file on the customer system before the package licenses can entered. This file could contain:

SysTec AG
3001;Danish Package
3002;English Demosystem
3003;German Package
A002;Voll
A005;Runtime
A00R;Nur lesen

The first line being the distributor name to be displayed in the license program, followed by the package numbers and names, and finally the usage codes.

7.4.2. Entering the distributor license

When a distributor is selected the license from this may the be entered as:



50. Two licensed packages from an external distributor

7.4.2.1. DISK input and output of license files

Like for TRIO it is possible to forward license files on disk. For distributor packages the .LIC and .PLS files will get the same name as for TRIO, but have a header line with the distributor number.
By DISK input of the .LIC file in the license program the correct distributor will be selected automatically, the customer does not have to use the menu first.
By DISK output of the .PLS file this information must of course be forwarded to the correct distributor by the customer.

7.5. Preparing a package on the distributor site

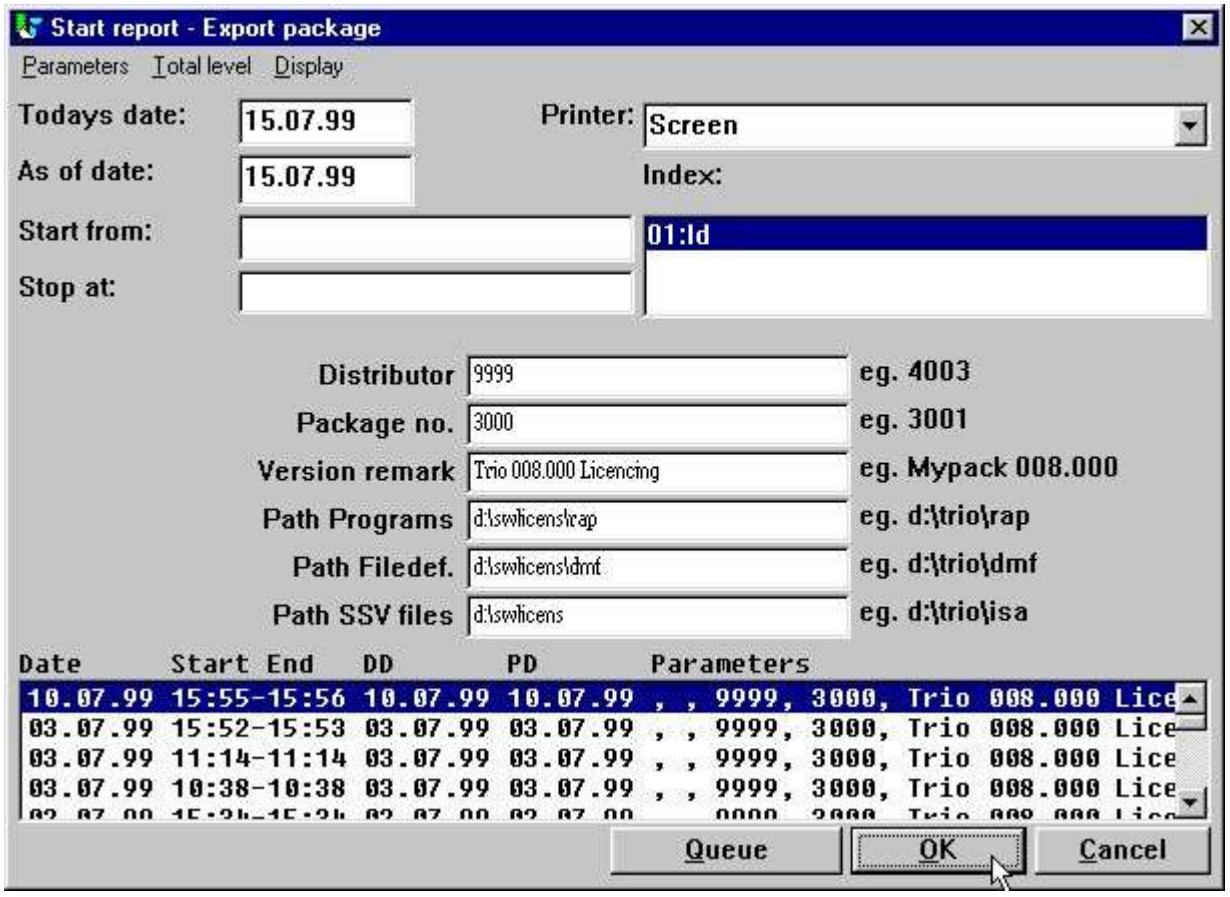
The package should normally be contained as source in one complete TRIO subsystem so the only thing to do is to zip the files together with the proper cryptation key and ship the zipfile.

7.5.1. The licensing package

A licensing package containing all necessary programs to enter and issue package licenses will be released for distributors, who needs to create own packages.
This set of TRIO programs contains all elements necessary to enter and issue the distributor licenses together with the package helpprograms.

7.5.2. Making the package ZIP file

In the distributor license package there is a report for this, scanning through the input files and calling the ZIP functions, which will deliver the proper cryptation key for the package number.



51. Starting the package report on distributor site

7.5.2.1. Contents of the package ZIP file

In the package ZIP file you may place any program definition DMxxxx.SRC / IQxxxx.SRC, program overviews DMREPORT.SSV / IQREPORT.SSV, file definitions xx.SSD and overview FILES.SSV.
Also pictures as for example LOGOs may be inserted.
For distributors making compiled reports, the DMxxxx.EXE compiled program may also be zipped, in which case the customer does not need a compiler to run this.

7.5.3. Issuing a license for the package

For the time being the license request must be forwarded to SW-Tools and the license registered here in a separate system until the license package programs is finally released and documented.
A distributor license with own logo may look like:

ASS + SySTeC
Stieghorster Strasse 115
D-33605 Bielefeld

Phone: 00-49-521-9202-100
Faxno: 00-49-521-9202-221
Email: MLudwig@systec-ag.de
www:

TRIO Distribution Incoorporated
Att.Mr.Good Salesman
On the Run 32
Byteaway 15
London SW1
England

Germany 04.07.99

Fax 0049 4711-0117

99914711 Seite 1 von 1

Wir bestätigen die folgende Installation:

Mit Wartung [X]

Lizenznr.	99914711
Vertreternr.	9991
Kunde Name	Data Coorporation Limited
Ihre Referenz	

Package license *4003 ASS + SySTeC*

Produkt	Verwendung	Datum	Bis	Lizenz Kode	Anzahl
3001 Package	5 Runtime	990704		**68UKPJ**	3
3002 Package	2 Voll	990704		**Y6FHHA**	3
3003 Package	3 Ligth	990704		**91TYZE**	3

52. A printed distributor package license

The logo printed is taken from the dddd.WMF file, here 4003.WMF.
The signature printed is taken from the XX.BMP file, XX being the first 2 letters of the user name entered in the license for the current PC.

7.5.4. Copying the package to the customer side

You will have to place the following files on the customer system:

- ddddllll.ZIP The package ZIP file
- LIddddPR.ENG The distributor license file
- nnnnnnnn.LIC The customer license file if possible

8. Other amendments and enhancements

8.1. Limits

The following numbers, which previously have been maximum 32.000 bytes, has been extended to 65.000 bytes on 16 Bit, 2.000.000.000 on 32 Bit version.

- **Recordlength of a file**
- **Length of one single alpha field and (from,to) specification**
- **Number of fields in one group (Tablefields)**
- **Total size of a report definition (DM1001.src)**

8.2. Memory and optimisation

The memory usage allows access from external DLL functions both on the 16 bit and 32 bit version to TRIO internal structures.
In general the modules was previously optimised mostly with respect to the 16 bit versions, now optimisations goes toward the 32 bit versions.

8.3. SW-Tools ODBC Driver

The ODBC driver has been extended to support long fileIDs, datadictionary calculations and external package file definitions also.

8.4. UNIX server SWTUSOCK

The default startup of the Unix server now uses the -x option causing each client to get his own process, which highly improves both performance and stability. To avoid using the -x option you may use -X (uppercase) to turn it off.

8.4.1. LINUX server

A version of SWTUSOCK for LINUX is available.

8.5. Descending index definition

The D options on field in index definitions may be used for the SSV and BASIC drivers and will be threaded exactly as I (invert all bits)

8.6. EXPORT Append to file

If a + is added to the export filename (as c:/tmp/ud+) data will be appended to the file instead of overwriting the file.

8.7. Japanese kanji codetables

The Japanese kanji codetables has been tuned, thanks to SNI Japan for this.

8.8. Network pathnames

It is now possible to use network computer names in the pathnames as
\\computername\sharename\directory .

8.9. User administration

A the time of this documentation the full user administration has not been updated for long fileIDs, basisIDs and fieldnumber modifications.

9. Subfunctions

The wellknown ZIP and UNZIP functions for file compression has been buildin and is the basis for PIP print archiving and, together with cryptation, separate licensed packages.
For printing, PIP and FORM has been added for archiving and forms.
GRID functions for list/combobox handling in IQ has been added.

9.1. The ZIP and UNZIP functions

9.1.1. UNZIP - Unpack files

Number UNZIP(Text *par1*, Text *par2*, Text *par3*, Number *par4*)

Par4: Max length of *par* 3

The UNZIP functions takes input from a ZIP compressed file made with the ZIP function or another product like PKZIP or WINZIP.

where the available options are:

```
-d              restore/create Directory structure stored in .ZIP file
-f              Freshen files in destination directory
-n              extract only Newer files
-o              Overwrite previously existing files
-s[pwd]         Decrypt with password
```

See also: ZIP

Example:

```
UNZIP("-d c:/tmp/zipfile")
UNZIP("d:/work/packfile","j:/tmp",#11,512)
```

9.1.2. ZIP - Zip file compression

Number ZIP(Text *par1*)

The ZIP functions compresses one or more files into a zipfile, which may later be unpacked using the UNZIP functions or other programs like PKZIP or WINZIP.

where the available options are:

```
    -a              Add files
    -b[drive]       create temp zipfile on alternative drive
    -c"comment"     set comment for all files
    -d              Delete files
    -e[x,n,f,s,0]   use [eXtra|Normal (default)|Fast|Super fast|NO compression]
    -f              Freshen files
    -i              add files with archive Attribute set (don't turn attribute
off)
    -m[f,u]         Move files [with Freshen | with Update]
    -o              set .ZIP file date to the latest file in .ZIP file
    -p|P            store Pathnames|p=recursed into|P=specified & recursed into
    -q              Quiet mode
    -r              Recurse subdirectories
    -s[pwd]         Scramble with password
    -t[date]        take files NEWER than or EQUAL to date (default=today)
    -T[date]        take files OLDER than date (default=today)
    -u              Update files
    -w|W<h,s>       include|exclude <Hidden, System> files (default=Whs)
    -x<file>        eXclude specified file
    -z"comment"     set .ZIP file header comment
```

See also: UNZIP,PIP

Example:

```
   ZIP("d:/work/files*.*")
```

9.2. PRINT functions

9.2.1. PIP - Print archiving

PIP(Fields *par1*,Filename *par2*,Text *par3*)

PIP enables you to control the print archiving for each report instead of for a printer in general.
Par1 should contain the fields to form the printindex file for search and display of printoverview, as invoice number, customer name or like.
Par2 contains the print archive filename, first char must be a letter. If no path is given the database path from the preference setup is used. If only 4 characters or less is given, YYMM is added to the filename and the date is inserted as 2.field in the printindex file.
Par3 can normally be omitted, in which case the print archives all pages printed since the last PIP function call. You may however state:

```
*        All pages printed
-        No pages, Just add entry to the SSV printindex
1-3,7    These pages
0        Current (Last) page only
```

0=OK, x=Failed.
See also: PRINT,ZIP
Example:

```
PIP("#1-3","kept")     /* Archive into keptYYMM with field 1-3 in printindex
```

9.2.2. SCRPRT - Recall screen print (IQ)

SCRPRT(Filename *par1*, Text *par2^)*

SCRPRT("filename") calls up the screen printer with the saved print from filename. This may for example be used in IQ by click on a field.
For the PIP Print archiving SCRPRT("filename,PIPID") is used, causing all files with the given PIPID to be unpacked from the filename.zip archive.
The second parameter may be used to tune the display on screen, it consist of a textstring with 5 values separated by comma:
Par2: "a,b,c,d,e"

```
a:   1  = Start display on page 1
b:   1  = Display in window,
     0  = Full screen
c:  -1  = Zoom factor, zoom out once to reduce size
d:   4  = If given, print on myprt 4, close after this is done
e: 3-4  = If d given, page range to print these pages only may be set
```

Returnvalue: None.
See also: PRINT,PIP
Example:

```
SCRPRT("c:/w/ab.cde")     /* Show this file using the screen printer
SCRPRT("c:/keepit/kept9908,7","1,1,-1") /* PIPID 7, page1 in window zoomed -1
```

9.2.3. **PRINT** - Print formula

PRINT(FORM=Text *par1*)
Parameters: *Par1*: Filename,WMF file for form

PRINT(FORM=c:/swtools/wmf/0101.wmf) defines a WMF picture file which will be used as background form when next page is printed.
Returnvalue: None.
See also: PRINT,PIP
Example:

```
PRINT(FORM=c:/swtools/wmf/0101.wmf)      /* Select this form
```

9.2.4. PRINT(LAB= - Label function (RAP)

PRINT(LAB=Text *par1*, Text *par2*, Text *par3*, Text *par4*, Text *par5*, Text *par6*)

par6 : Copies

Description: The width and height of any label on the sheet can be given in centimetres or inches by using the following syntax:

```
7cm equals 7 centimetres
2in equals 2 inches
```

The below sample produces labels printed from left to right on a label sheet with 21 labels, 3 on each row, 7 rows, where each label has the width/height of 7 centimetres. Each label is printed in 2 copies.

Returnvalue: None.

See also: PRINT

Example:

```
FIRST
PRINT(LAB=1,3,7,7cm,7cm,2) /* Define label print
NORMAL
```

9.3. The GRID Functions for IQ

9.3.1. DGRID - Dialog Database Grid (IQ)

DGRID(text *par1*, text *par2*)

Description: This function may be used to generate a dialog of records from a selected table. The dialog will display the records according to an existing index and the fields requested.
The syntax of the definition in *par1* is:

A!B!C!D!E!F!G!H!I!J!K

where each parameter is separated by the character "!".

A - File id
B - Index
C - Fields to display
D - Column to return value from (Origin 0)
E - Dialog header
F - Selections (Optional - Refer to EXEC)
G - Reserved
H - Reserved
I - Reserved
J - Screen x position (optional)
K - Screen y position (optional)

Please note the returnvalue is set to origin 0 of the fields requested to be displayed. For example, if defined

va!1!1-6!0!Select an article

table va is opened using index 1 and display of fields 1-6. If a row is selected the returned value will be the value of display field 0, e.g. field 1 (Article number).
If defined

va!2!1,2,6!2!Select an article

table va is opened using index 2 and display of fields 1, 2 and 6. If a row is selected the returned value will be the value of display field 2, e.g. field 6 because

Display field 0 = field 1
Display field 1 = field 2
Display field 2 = field 6

If required to control which records are to be displayed, it may be required to perform selections. As TRIO already supports the command EXEC, it is possible to use a parameter F for this:

va!1!1-6!0!Articles in group 9!va#7==9

This will only include articles within article group 9.
Returnvalue: 0=Ok, row selected x=No row selected, Dialog closed
Example: DGRID("va!1!1-6!0!Select an article",#50) /* *User may select an article*

9.3.2. GRIDHDR - Set header for grid (IQ)

GRIDHDR(fields *par1*, fields *par2*, Bitflag *par3*)

32768 = Line pointer active

Description: The heading may be set by reference to fields whereas the field names will be used. For example,

va#1-6

will generate a heading as

Article no . Name . Selling price .....

The format of each field will determine if the column is left or right justified, but may be changed with an option after the field reference. Also, a field may be inserted into the grid without being displayed or scrolled. The following options may be used:

l - left justified
r - right justified
h - hidden
s - no scroll
i - no input

A definition such as

va#1s,2,6h,7l

will display field 1 as a no scroll field, field 2, field 6 will not be displayed and field 7 will be left justified.

Returnvalue: None.

See also: GRIDFLD

Example: GRIDHDR("#50","va#1-10") */* This defined the grid header as field 1-10 from the article table*

9.3.3. **GRIDFLD** - Set fields values into grid row (IQ)

GRIDFLD(fields *par1*, fields *par2*, number *par3*)

par3 : Row number
Description: The function inserts the value of fields in *par2* into the grid. Please remember manually to calculate the row number for *par3* which starts from 0.
Returnvalue: None.
See also: GRIDHDR
Example: GRIDFLD("#50","va#1-10",0) /* *Insert fields values in first row*

Figur liste

Index