



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

TRIO Release Information (007.000)

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

Contents

Contents	2
1. Introduction	5
1.1. TRIO 32 Bit release	6
1.2. Charts	7
1.3. OLE 2.0 support.....	8
1.4. USER administration.....	9
1.4.1. Report start statistics and jobqueues	10
2. Installation	11
2.1. Running TRIO from the CD.....	12
2.2. Selecting TRIO components to be installed.....	13
2.2.1. Installing a UNIX server using FTP.....	14
2.3. Server programs.....	15
2.4. Version number	16
2.5. Licensing	17
3. The Data Dictionary	18
3.1. Number of fields increased to 999	19
3.2. Floating decimals	20
3.3. Thousand editing	21
3.4. Index key definitions, LOGICAL index	22
3.4.1. Directory 00 recordnumber reading	23
3.4.1.1. ACCESS using SWODBC on non-indexed files.....	24
3.4.2. Suppressing IQ search on specific index (*xx)	25
3.4.3. Suppressing index update for a specific index (-xx)	26
3.4.4. Creating a pseudoindex for a file (+xx)	27
3.4.5. Index example.....	28
3.4.6. Key options, Descending index.....	29
3.5. File connection definitions	30
3.5.1. Multiple connections to the same file using upper/lower case.....	31
3.5.2. Multiple connections to the same file using logical index.....	32
3.5.3. Forcing connections by READ.....	33
3.6. Packtypes for fields	34
3.6.1. Secondary packtypes.....	35
3.6.1.1. Secondary packtype parameters	36
3.6.1.2. Default secondary packing for BASIC dates.....	37
4. Year 2000.....	38
4.1. Preferences	39
4.1.1. Date output format.....	40
4.1.2. Internal date calculation format	41
4.2. Server and file update	42
4.3. Date sorting	43
4.4. DOS dates	44
4.5. Date function updates	45
4.6. Year 2001 problem.....	46
4.7. Year 2005 problem.....	47
4.8. Year 2010 problem.....	48
4.9. Year 2100 problem.....	49
4.10. Date input validation	50
4.10.1. IQ selection date input.....	51
4.10.2. CHAIN function masked date input	52
5. Database interface drivers	53
5.1. BASIC Interfaces	54
5.2. <u>NETBASIC</u> using CTRAS functions	55

5.3. <u>Quattro</u> interface using CTRAS.....	56
5.4. <u>Quattro</u> interface using 32 Bit FTP.....	57
5.4.1. Working offline with the Quattro files.....	58
5.4.2. File definitions	59
5.4.3. <u>XBasic</u> and other interfaces using 32 Bit FTP.....	60
5.5. <u>GLOBAL</u> 3000 Speedbase and Global 2000 ISAM/DMAM.....	61
5.6. <u>NAVISION</u> Financials	62
5.6.1. Install Navision Financials and C/FRONT.....	63
5.6.2. Install and configure the Navision Financials interface.....	64
5.6.3. Import of table definitions	65
5.7. <u>CONCORDE</u> C5/XAL.....	66
5.7.1. Exporting the Concorde file definitions	67
5.7.2. Install and configure interface to Concorde.....	68
5.7.3. Import of table definitions	69
5.8. <u>Btrieve</u> version 5.10 and 6.15.....	70
5.8.1. Install and configure interface to Btrieve	71
5.8.1.1. Advanced options	72
5.8.1.1.1. Open mode.....	73
5.8.1.1.2. Name suffix	74
5.8.1.1.3. Name options.....	75
5.8.1.1.4. Parameters	76
5.9. <u>ODBC</u> drivers interface - Application type.....	77
5.9.1. Type 1 - Support of Informix Database table locks	78
5.9.2. Type 4 - Always perform ORDER BY	79
5.9.3. Type 4096 - Alpha fields may NOT be NULL.....	80
5.9.4. Type 8192 - ODBC Drivers with support of one connection per table only	81
5.10. <u>ISAM</u> database interface - Application type.....	82
5.10.1. Type 2 - Support of SAMSON system	83
5.11. <u>SSV</u> database interface	84
5.11.1. Write sortindex	85
5.11.2. Headerlength	86
5.11.3. First data record.....	87
5.11.4. Make indexfile > records	88
5.11.5. Write CRLF	89
5.11.6. Indexfile naming	90
5.11.7. Duplicates >= index	91
5.11.8. Read CRLF.....	92
5.11.9. Compression used	93
5.11.10. Initial codetable.....	94
5.12. <u>Extended</u> SSV interface.....	95
5.12.1. Fixed recordlength.....	97
5.12.2. Fixed blocklength	98
5.12.2.1. Max no. of data records.....	99
5.12.3. Fixed records/block	100
5.12.4. First free record	101
5.12.5. No.of free records.....	102
5.12.6. Next free record	103
5.12.7. Filename type	104
5.12.8. Field Pack types	105
5.12.9. Key Pack types.....	106
6. Printing	107
6.1. Screen printer enhancements	108
6.1.1. Saving the screen print to a file	109
6.1.1.1. SCRPRT function for IQ print recall	110
6.2. Multi Output Format printers	111

6.2.1. Printer type	112
6.3. Text printer, Printertype 1	113
6.3.1. Options	114
6.4. HTML printer, Printertype 2	115
6.4.1. HTML output of standard reports	116
6.4.2. HTML output of non-standard reports	117
6.4.3. Background, logo and browse buttons	118
6.5. RTF printer, Printertype 3	119
6.6. TXT printer, Printertype 4	120
6.7. SSV printer, Printertype 5	121
6.8. Options for printer type 2 - 5.....	122
6.8.1. Output to file.....	123
6.8.2. View output with	124
6.8.3. Definition file	125
6.9. Printing Labels.....	126
6.9.1. How to use in RAPGEN.....	127
6.10. RAPGEN start parameter	128
7. Calculations	129
7.1. Wizard function helper.....	130
7.2. External DLL functions.....	132
7.3. <u>DIALOG</u> function for additional input	133
7.4. <u>PARAMS</u> function for additional report start parameters	134
8. Subfunctions.....	136
8.1. <u>DIALOG</u> - Function for additional input.....	137
8.2. <u>PARAMS</u> - Function for additional report start parameters (RAP)	138
8.3. <u>FTP</u> - File Transfer Processor	139
8.4. <u>GETINFO</u> - Get additional program information (IQ/DM)	140
8.5. <u>LINE</u> - Retrieve or set the current line number (IQ/DM)	141
8.6. <u>OBJECTGETSTRING</u> - Get index of an objects selected item (IQ/DM)	142
8.7. <u>PRINT(LAB=</u> - Label function (RAP)	143
8.8. <u>SCRPRT</u> - Recall screen print (IQ)	144
8.9. <u>SHOW</u> - Enable/Disable/Show/Hide a field (IQ/DM)	145
8.10. <u>USERINFO</u> - Get information about user	146
Figure list.....	147
Index.....	148

1. Introduction

The release 007 of TRIO contains many improvements such as 32 Bit version, Charts, OLE, FTP, additional database interfaces, user administration and function help wizard. For customers with maintenance contract the additional functionality is free of charge, however upgrading to 32 Bit must be paid as is the User Administration module including report statistics and jobqueue considered as a separate application.

1.1. TRIO 32 Bit release

All modules are now available in 32 Bits versions for Windows 95 and NT. The 32 bits version will work exactly like the 16 bits versions apart from a speed improvement which has been measured to between 10 to 75 % dependent on the user program.

On many installations the requirement of servers will change as the 32 Bit Windows server may no longer be needed on NT X-Basic systems which also should reduce the overhead. Quattro customers using 16 Bit Ctras must stick to TRIO 16 bit as CTRAS will not be released for 32 Bits.

1.2. Charts

The chart functions provide the user with a very simple way to produce a chart for report just by checking this option on the total dialogbox. Also an IQ program may easily be extended with a chart.

Different chart forms are available as Bar, Pie, Line and XYZ point and a set of calculation functions are added for the advanced users who needs an extensive control of chart options.

The TRIO Charts manual describes these functions in details.

1.3. OLE 2.0 support

Support of OLE 2.0 Object Linking and Embedding gives the user possibility of integrating other Windows applications in the report or query defined in TRIO.

For example Microsoft Word may be used when defining a customer letter and a video sequence may be played on an article query.

The interface is simple to use as the OLE object is just defined as a field in TRIO marked as OLE and placed on the report form / IQ query just like a picturefield. Display and amendment of the contents will then be carried out by the selected OLE server program. Optionally a link to an existing document may be used or an object embedded and stored within the TRIO program can be created.

The TRIO OLE manual describes these functions in details.

1.4. USER administration

By means of the User administration it is possible to restrict the set of reports / IQ programs and files which are available to the enduser. Permissions may be given as overall for a TRIO module or specified detailed for single fields in the files for read or write.

Users may be grouped together with same permissions or quite individual systems may be created for one or more users overhauling some of the subsystem control used today.

The TRIO USER Administration manual describes these functions in details.

1.4.1. Report start statistics and jobqueues

A logfile with the last 100 start informations is kept for each report and displayed by report start. The parameters may be reused for next start or an average runtime statistic may be shown just by click on an entry line.

Queues of reports may be builded which may then be started just as a single report.

2. Installation

TRIO is now distributed on CD-Rom and the autorun facility has been added so installation screen will be shown automatically when the CD is mounted on a Windows 95 system.



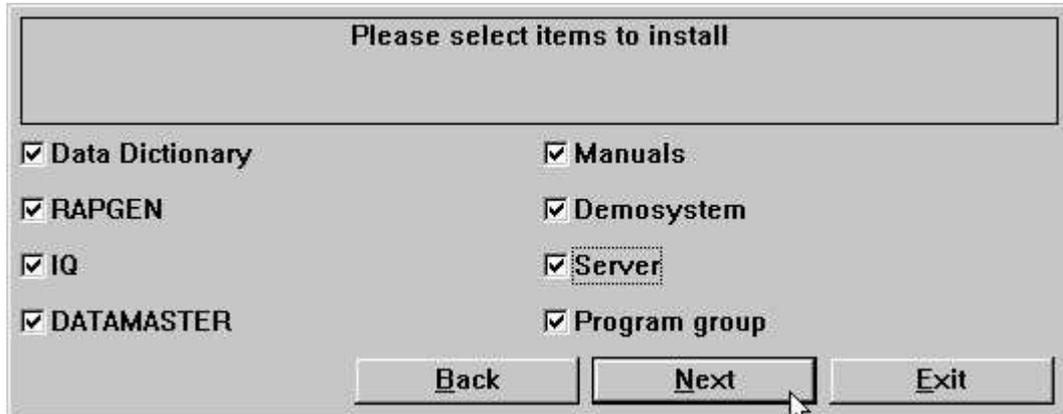
1. The CDMENU selection

2.1. Running TRIO from the CD

If you select the TRIO-CD installation a demo system will be created on the harddisk using a minimum of disk space. All programs and help manuals will be taken from the CD.

2.2. Selecting TRIO components to be installed

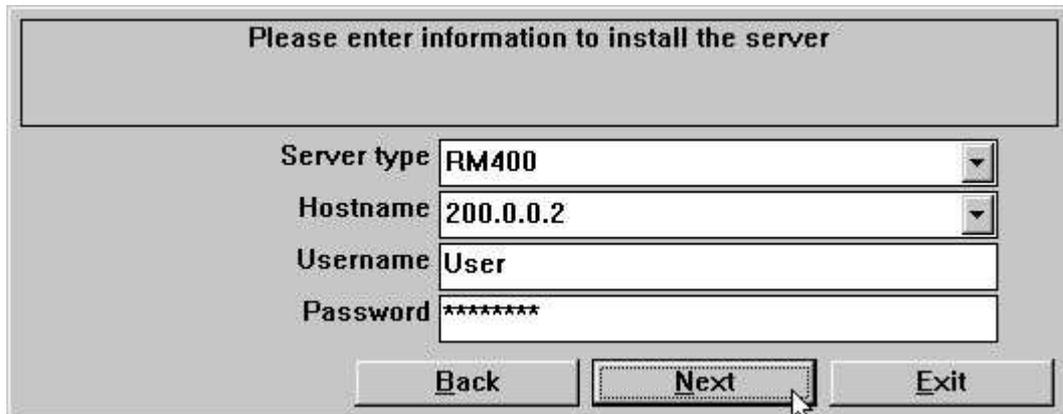
When TRIO installation is selected you have the choice to install one or more of the components:



2. The installation selection

2.2.1. Installing a UNIX server using FTP

The installation now has builtin FTP access for easy transfer of the server program onto the Unix system.



Please enter information to install the server

Server type RM400

Hostname 200.0.0.2

Username User

Password *****

Back Next Exit

3. The FTP specification by server installation

2.3. Server programs

Previous versions of TRIO will run with the new server as well will the new release run with old servers. However we recommend the versions matches.

The new server of course includes support for the new interfaces and must be installed where these is to be used. Also the stability and speed of the new server program has been improved by use of blocking of network packages.

For OS/2 client systems multiple processes on TCPIP network could hang up the server. This problem is now solved.

2.4. Version number

The TRIO version number is found in the HELP-ABOUT menu or on any print of documentation from TRIO.



4. TRIO Version number

As shown the 32 bit version states 007.3xx as 300 is added to the minor version number.

2.5. Licensing

TRIO 32 bit requires own license codes.

User administration requires special license code.

Without maintenance you will need an upgrade version code to use TRIO 007.xxx if installation is before 01/08-1998.

A licensecode issued for TRIO version 007 may be used with version 006 also. You may have to update the product table contained in the textfile LICENSPR.SSV as all product codes must be known to the license program.

3. The Data Dictionary

Minor enhancements to the field format has been done as described below, and some new packtypes has been implemented due to support of new database systems.

Logical index definitions has been introduced to allow definition of pseudokeys for non-indexed files and the READ has been reworked for this.

The year 2000 triggers the additional secondary packing (calculation) on fields.

3.1. Number of fields increased to 999

The number of fields in one filedefinition has been increased from 499 to 999.

3.2. Floating decimals

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

3.3. Thousand editing

Request of thousand editing of 6 or 8 digit fields has been honoured as:

,6, or ,8, gives date fields (,4, is reserved for time)

,6,0 and ,8,0 and ,4,0 may be used for 999.999 / 99.999.999 / 9.999

3.4. Index key definitions, LOGICAL index

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

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

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

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

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

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

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

3.4.1. Directory 00 recordnumber reading

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

3.4.1.1. ACCESS using SWODBC on non-indexed files

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

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

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

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

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

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

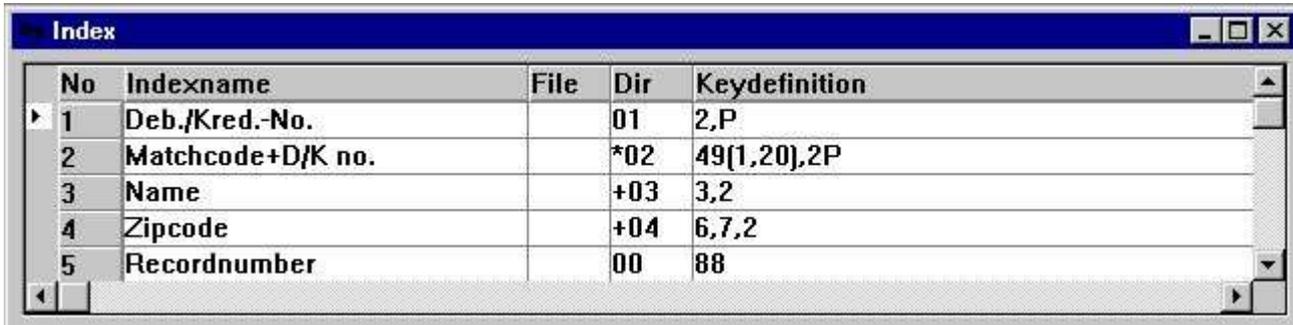
- THIS FUNCTION IS FOR THE TIME BEING NOT FINALLY RELEASED -

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

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

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

3.4.5. Index example



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

5. Example of pseudoindex and recordnumber definition

3.4.6. Key options, Descending index

A couple of keyfield options has been added:

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

C = Global files only, Compress DMAM index

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

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

3.5. File connection definitions

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

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

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

READ(xx) searches for a connection to file xx stated in lowercase first, if not present the first connection found is used regardless of case. XX, Xx, xX may be used to state different connections. The sorting of file connections in the FDF has been changed so all connections to a file is shown together regardless of case.

3.5.2. Multiple connections to the same file using logical index

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

3.5.3. Forcing connections by READ

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

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

3.6. Packtypes for fields

The following field packtypes has been added:

Type	Name	Is stored as	Usage
1053	LDCHARR	Right justified field no extra byte as 1043	C-Isam
1121	NAVI_STR	Textstring	Navision P5
1122	NAVI_DATE	Date structure	Navision P6
1123	NAVI_TIME	Time structure	Navision P3
1124	NAVI_BLOB	Binary large object	Navision
1125	NAVI_BOOL	Boolean	Navision P2
1126	NAVI_S16	Short	Navision P7
1127	NAVI_S32	Long	Navision P8
1128	NAVI_U8	Char (byte)	Navision P1
1129	NAVI_ALFA	Textstring with 1.byte=Length	Navision P
1130	NAVI_BCD	Amount	Navision P9
1131	NAVI_O32	Optioncode	Navision P4

3.6.1. Secondary packtypes

Due to year 2000 updates it has been necessary to introduce a secondary packtype which can be seen as a fixed calculation performed just after the field has been read and unpacked according to the Pxxxx packtype, or just before the field will be packed and written to the record. The secondary packtype is given as Vxx within the field format as: ,6,V1

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

3.6.1.1. Secondary packtype parameters

A parameter may be given together with the secondary packtypes as: ,6,V1.30 This may be used to overrule the default year wrapping by converting to/from short date formats.

3.6.1.2. Default secondary packing for BASIC dates

For short date fields (,6,) in BASIC files the secondary packtype will default to V for numeric fields, V3 for alphafields and V4 for packed fields. These rules are also applied to any keyfields containing short dates.

4. Year 2000

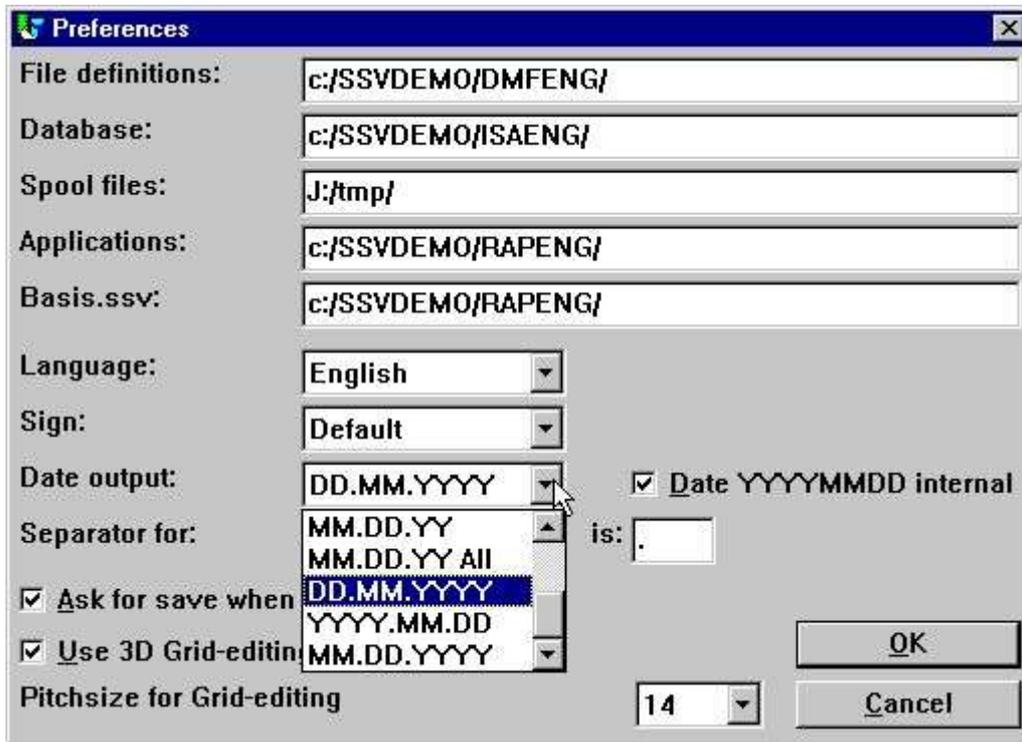
As TRIO is integrating to many different file systems all sort of year 2000 patching and tricking are seen.

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

4.1. Preferences

The preference setup for TRIO now includes a X-mark defining all dates being internally handled as YYYYMMDD independent of how these should be stored in the files.

This enables writing of calculations independent of the sometimes complex conversion rules made to solve the year 2000 problem in the file systems.



6. Preferences

4.1.1. Date output format

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

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

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

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

4.1.2. Internal date calculation format

- BE CAREFULL BEFORE YOU ACTIVATE THIS OPTION -

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

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

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

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

You will have to change your calculations after activating this:

IF #47>#DD LET #99=1

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

4.2. Server and file update

You must observe that writing to files will require server version 007.001 if the index keys are to be constructed with shortdate conversion.

4.3. Date sorting

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

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

4.4. DOS dates

DOS and equivalent systems will return 100 for year 2000.
A date field of ,6, or ,8, with value between 1.000.000 and 1.999.999 will be assumed to be in year 2000.

4.5. Date function updates

All the date functions has been proved to work with year 2000.

The buildin calendar for the WORKD function has been extended until year 2010, please remember to update your individual table if you use workday calculations.

4.6. Year 2001 problem

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

FNY(31.03.98) = 1998.03.31 ok

FNY(31.03.00) = 2000.03.31 ok

FNY(31.03.01) = 2031.03.01 YYMMDD is preferred instead of DDMMYY

4.7. Year 2005 problem

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

4.8. Year 2010 problem

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

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

4.9. Year 2100 problem

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

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

4.10. Date input validation

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

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

4.10.1. IQ selection date input

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

4.10.2. CHAIN function masked date input

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

5. Database interface drivers

The number of database interfaces for TRIO is growing and extending the possibilities for the user to mix or convert from/to many platforms.

5.1. BASIC Interfaces

The following lists the available BASIC interfaces:

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

5.2. NETBASIC using CTRAS functions

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

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

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

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

5.3. Quattro interface using CTRAS

Note that the CTRAS library functions for Quattro are 16 bits only and will not be released as 32 bits in the near future.

Due to different problems with the interface no 16 Bits Windows server will be released with TRIO either. Instead the FTP Quattro interface as described below may be used with TRIO 32 Bits.

5.4. Quattro interface using 32 Bit FTP

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

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

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

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

7. Setting up Quattro using FTP

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

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

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

5.4.1. Working offline with the Quattro files

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

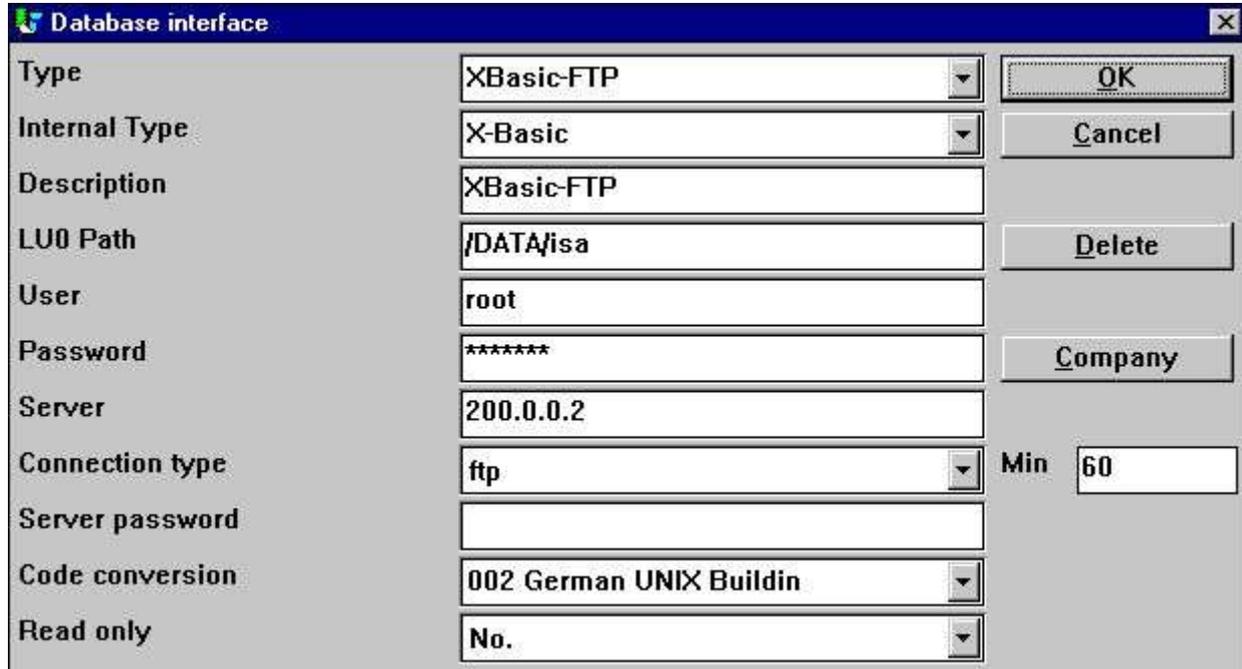
If a file cannot be transferred due to connection is not available an error message is given. If an earlier transferred file is available the program will use data from this without new transfer. You may transfer files manually to the PC from a quattro system also using the FTP (see this) or tape transfer.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

5.4.2. File definitions

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

5.4.3. **XBasic** and other interfaces using 32 Bit FTP

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



8. Setting up XBasic using FTP

However one path only is available which means the local path will become the same as the path on the server. The local path will be created if not present. By the way you should note that the XBasic interface, also in CTRAS-Library write mode, may be used on an UNIX system even if XBasic itself is not installed.

5.5. GLOBAL 3000 Speedbase and Global 2000 ISAM/DMAM

TRIO now supports these systems. A separate installation manual is available on request.

5.6. NAVISION Financials

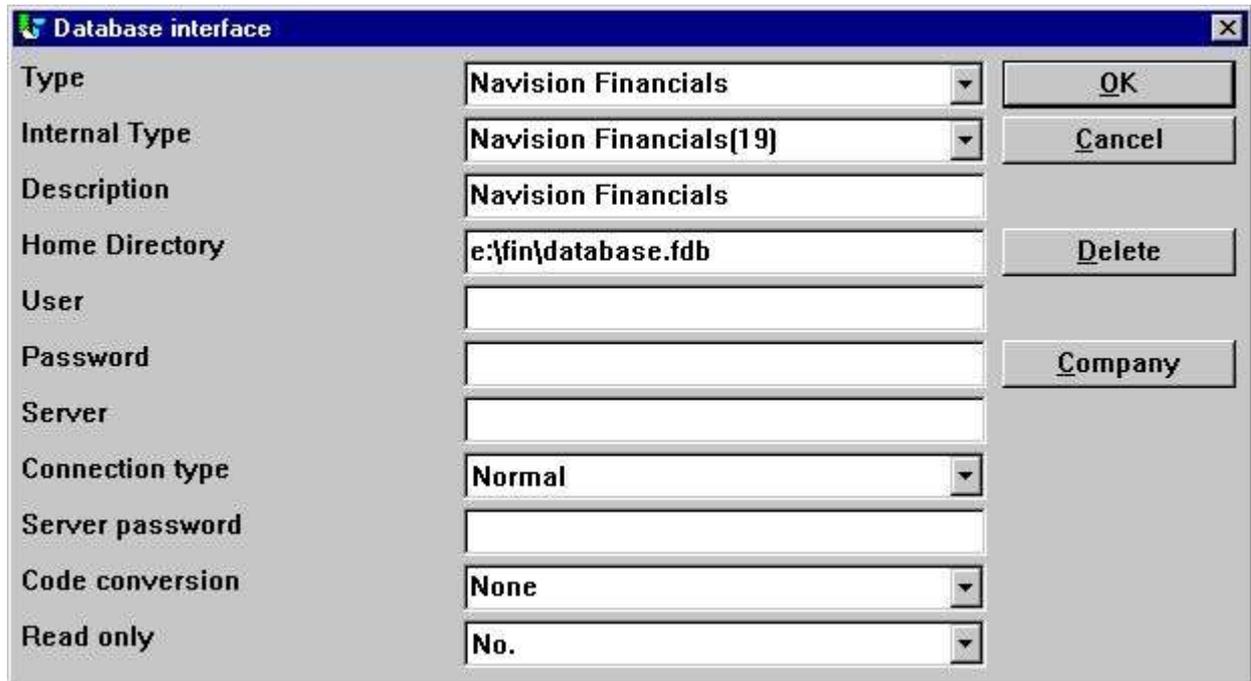
The 32 bit version of TRIO and the ODBC driver now interfaces the Navision Financials package by use of the C-Front module which must be present on the system. Both read and write are implemented, C/FRONT version 1.2 or higher is supported.

5.6.1. Install Navision Financials and C/FRONT

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

5.6.2. Install and configure the Navision Financials interface

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



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

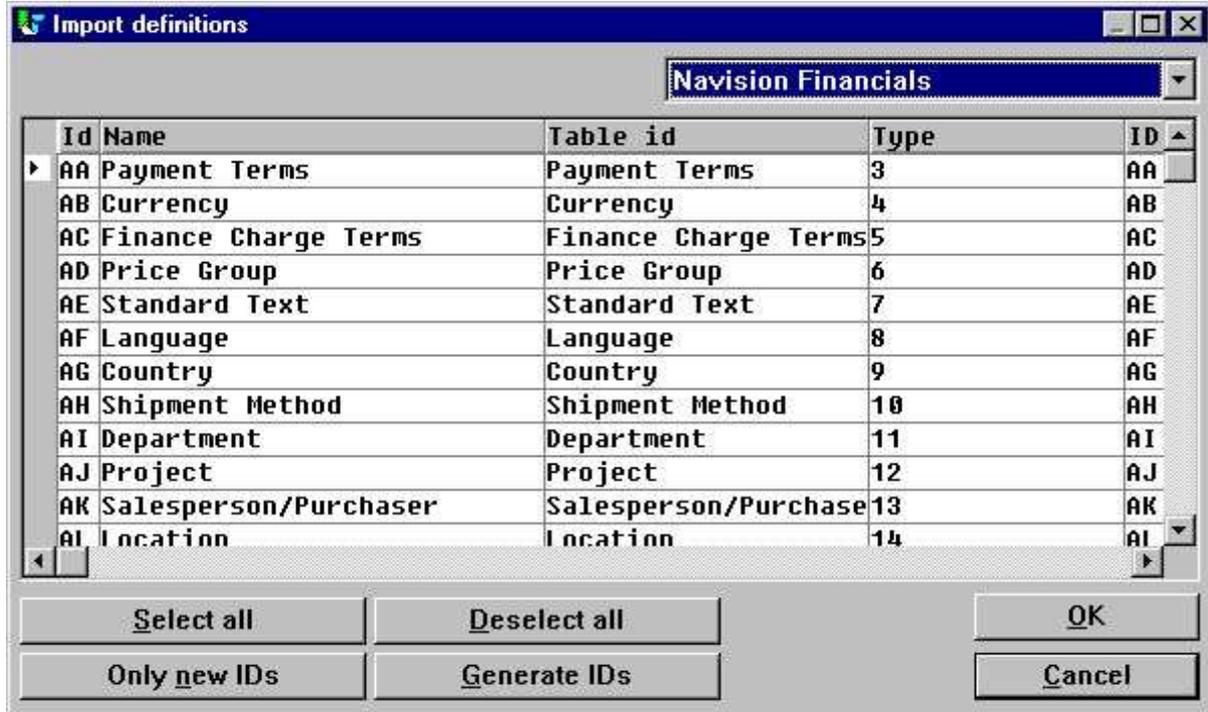
Field	Value	Action
Type	Navision Financials	OK
Internal Type	Navision Financials(19)	Cancel
Description	Navision Financials	
Home Directory	e:\fin\database.fdb	Delete
User		
Password		Company
Server		
Connection type	Normal	
Server password		
Code conversion	None	
Read only	No.	

9. Sample setup of Navision Financials interface

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

5.6.3. Import of table definitions

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



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

5.7. CONCORDE C5/XAL

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

5.7.1. Exporting the Concorde file definitions

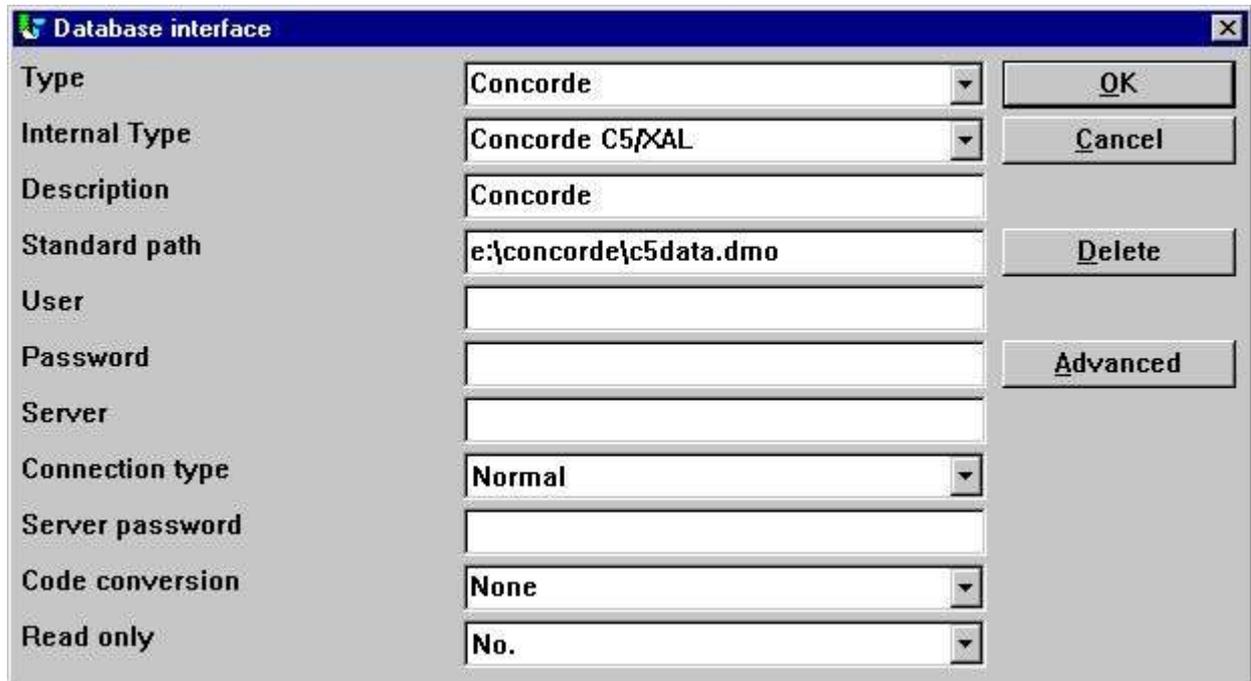
If Concorde is installed for example in directory e:\concorde you should first export all file definitions into a .DBD file.

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

5.7.2. Install and configure interface to Concorde

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

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



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

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

11. Sample setup of Concorde interface

5.7.3. Import of table definitions

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

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

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

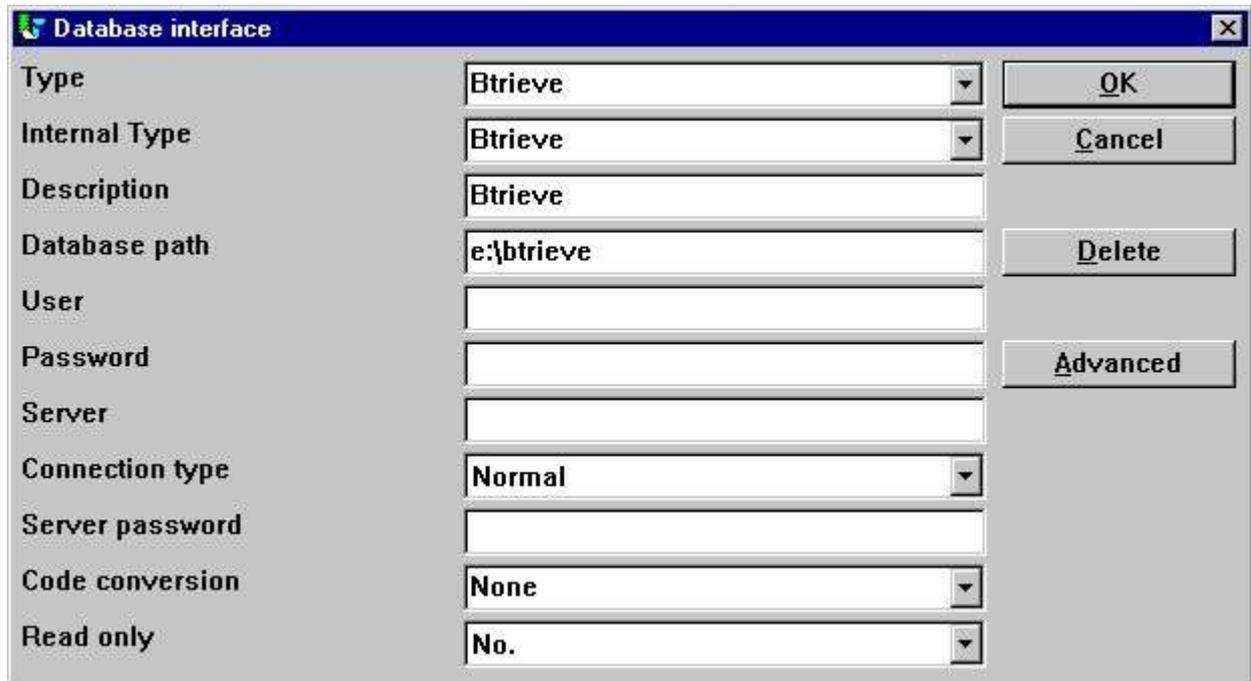
5.8. Btrieve version 5.10 and 6.15

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

5.8.1. Install and configure interface to Btrieve

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

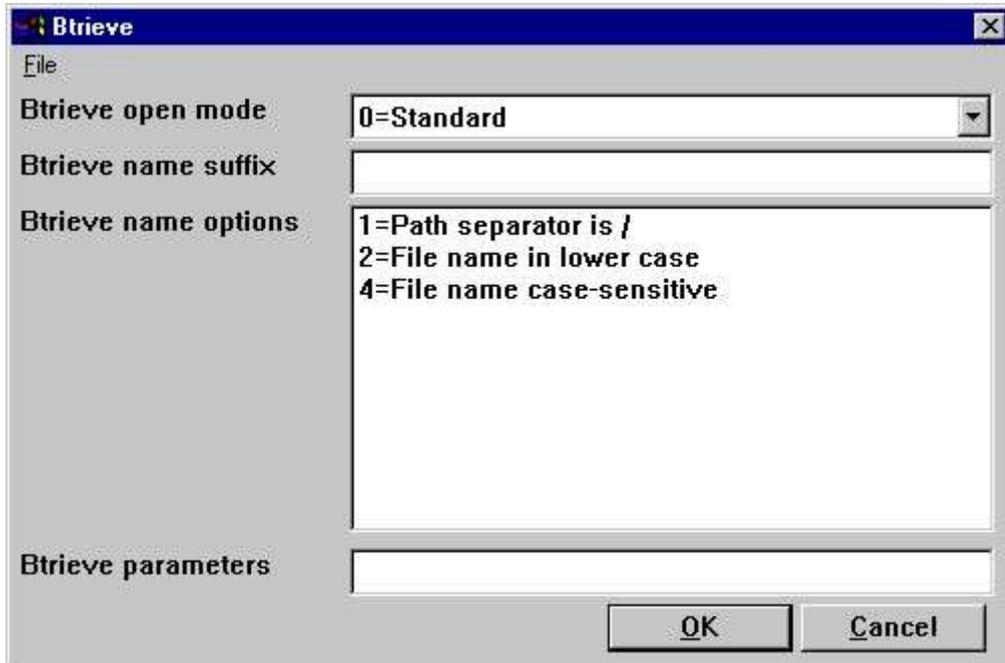
By the function 'Database-interface' you may select the installed 'Btrieve' where the following setup will correspond to access of Btrieve files located in the directory 'e:\btrieve'.



12. Sample setup of Btrieve interface

5.8.1.1. Advanced options

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



13. Advanced settings on Btrieve interface

5.8.1.1.1. Open mode

The open mode may be one of the following:

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

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

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

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

5.8.1.1.2. Name suffix

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

5.8.1.1.3. Name options

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

c:/btrieve/database

and a file is defined with name

customer

and the suffix is set as

dat

the actual location of the file is

c:/btrieve/database/customer.dat

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

c:\btrieve\database\customer.dat

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

5.8.1.1.4. Parameters

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

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

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

5.9. ODBC drivers interface - Application type

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

5.9.1. Type 1 - Support of Informix Database table locks

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

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

5.9.2. Type 4 - Always perform ORDER BY

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

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

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

If type 4 is set it will generate

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

5.9.3. Type 4096 - Alpha fields may NOT be NULL

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

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

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

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

5.10. ISAM database interface - Application type

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

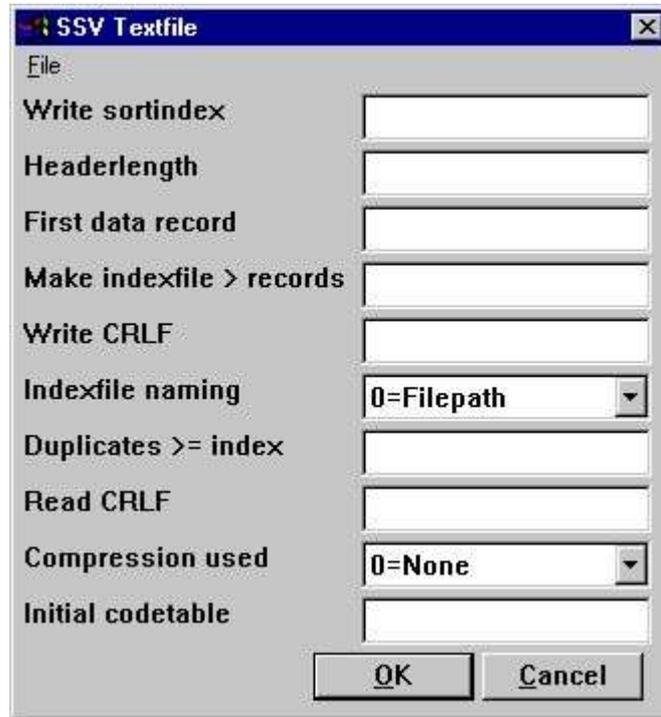
5.10.1. Type 2 - Support of SAMSON system

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

5.11. **SSV** database interface

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

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



14. The ADVANCED options with the SSV driver

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

5.11.1. Write sortindex

Determines sorting by write of the file.

0 = No sorting, recordnumbers (lines) are kept

>0 = Sort the file according to this index definition

5.11.2. Headerlength

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

5.11.3. First data record

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

First real data position is calculated as: Headerlength + First data record * Recordlength

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

512 Decimal
0x200 Hexadecimal
***8,B16P1006**

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

5.11.4. Make indexfile > records

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

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

5.11.5. Write CRLF

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

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

5.11.6. Indexfile naming

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

0 = filepath/filename.Xnn

1 = TMP/filename.Xnn

5.11.7. Duplicates >= index

Duplicates will normally not be allowed in an index.

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

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

5.11.8. Read CRLF

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

5.11.9. Compression used

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

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

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

1 = May be compressed

5.11.10. Initial codetable

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

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

5.12. Extended SSV interface

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

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

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

The screenshot shows a window titled 'File' with the following fields and controls:

- File ID:** A dropdown menu showing '54 Customerfile'.
- SSV-Extended:** A dropdown menu.
- File name:** A text box containing 'gf-03000001.'
- Standard ID:** A text box.
- Real name:** A text box containing 'c:/DATA/isa/gf-03000001.'
- GF-03000 4 JH:** A text box.
- File text:** A text box containing 'Customerfile'.
- Documentation:** A button.
- Advanced:** A button with a mouse cursor pointing to it.

15. The PARAMETER button for single files.

The screenshot shows a window titled 'Kunden-Stammdatei' with the following parameters:

Write sort index	0	Fixed recordlength	*8,B2P1005*2
Headerlength	256	Fixed blocklength	
First data record	*8,B16P1006	Max no of data records	*8,B8P1006-*8,B16P
Make indexfile > records		Fixed records/block	
Write CRLF		First free record	*8,B20P1006
Indexfile naming	0=Filepath	No.of free records	*8,B24P1006
Duplicates >= index		Next free record	*8,B0P1001
Read CRLF		Filename type	0
Compression used	0=None		
Initial codetable	49		
Field Pack types	1P1100 1,P1101 1,P1		
Key Pack types	1P1100 1,P1101 1,P1		

Buttons: OK, Abbrechen

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

5.12.1. Fixed recordlength

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

***8,B2P1005*2**

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

5.12.2. Fixed blocklength

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

***8,B2P1005*2**

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

5.12.2.1. Max no. of data records

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

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

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

5.12.3. Fixed records/block

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

5.12.4. First free record

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

***8,B20P1006**

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

5.12.5. No.of free records

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

***8,B24P1006**

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

5.12.6. Next free record

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

***8,BOP1001**

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

5.12.7. Filename type

Reserved for future use, leave this field as 0.

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

5.12.8. Field Pack types

May be used by fixedlength files only, a normal SSV file always uses SSV packing.
The packtypes may be given directly on the fields as e.g. P1013, or the default may be given here for each fieldtype separated by blank.

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

.
gives the normal BASIC packing of Alpha, 1 word, 2word, ... , Call 60 etc.
The first 6 being for fieldtypes 0-5 followed by P0-9

5.12.9. Key Pack types

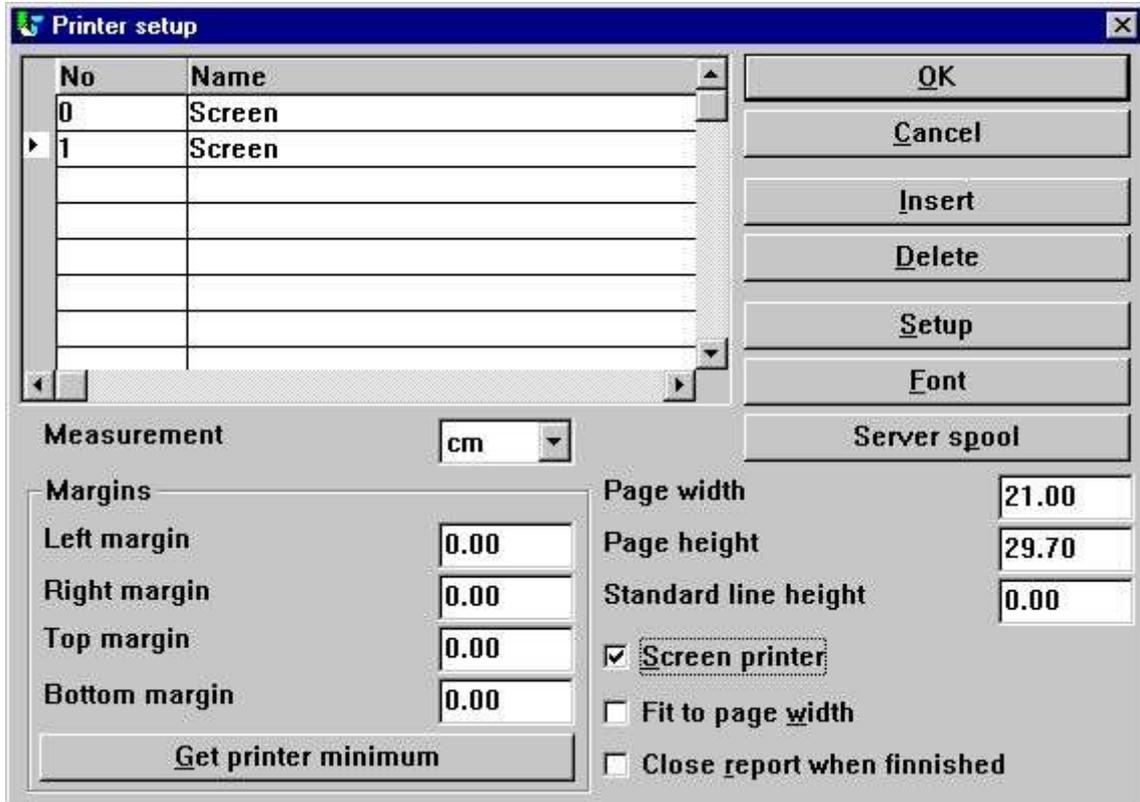
As fields packtypes but used when a field is used in a key, se above.

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

6. Printing

6.1. Screen printer enhancements

The screen setup in TRIO may now be given multiple time by setting the check mark for screen printer on an empty printer line in the setup dialog.

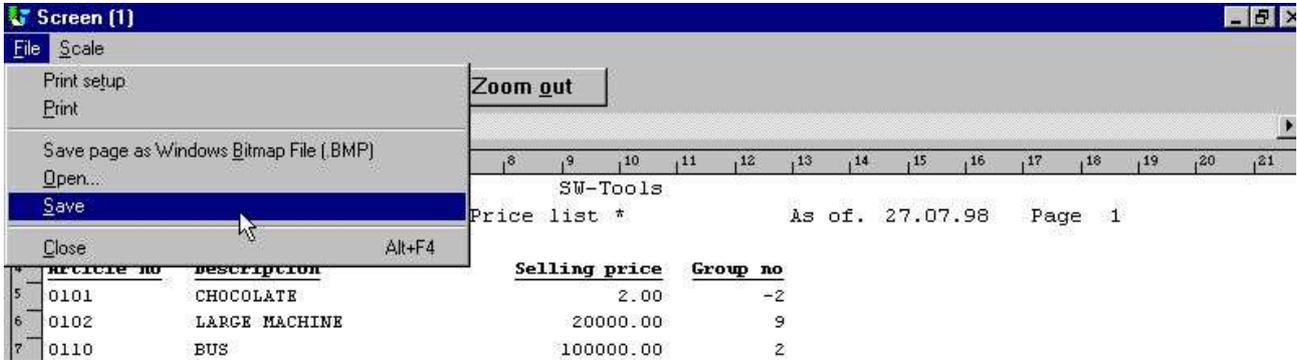


17. Additional screen printer insertion

Different screenprinters may be setup with different pagesize and as you will see below may be used for Multi Output Format to textfiles, HTML and RTF also.

6.1.1. Saving the screen print to a file

In the file menu for the screen printer display a possibility has been added to save a print to a file and to recall this later.



18. Saving a screen print to a file

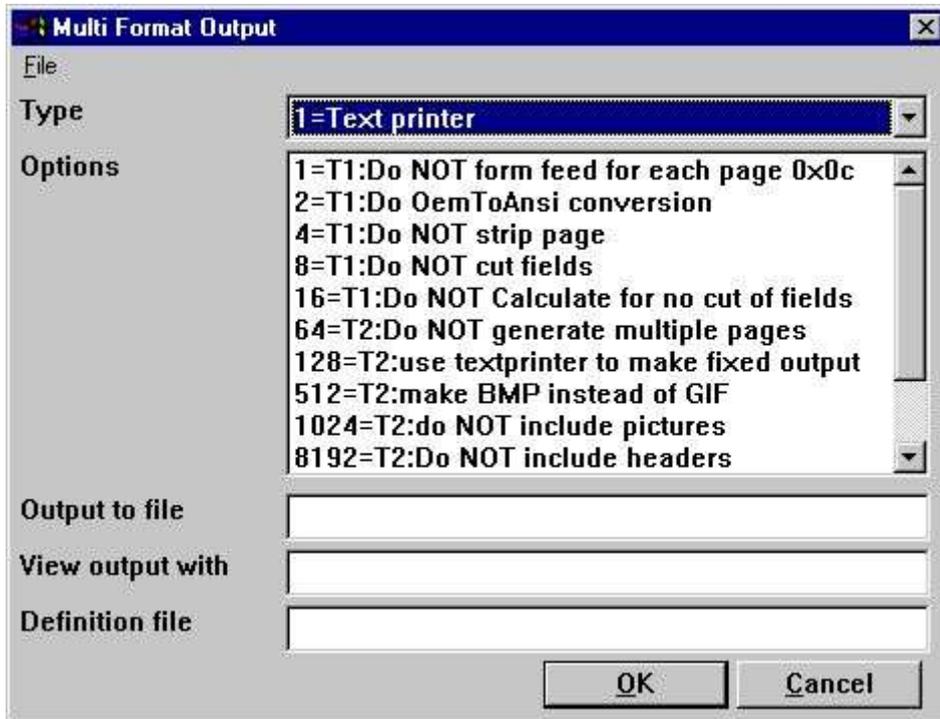
However you should note that special print as charts and OLE linked objects will not be included into to saved print.

6.1.1.1. SCRVRT function for IQ print recall

The SCRVRT("filename") function in IQ may also be used to recall the print.

6.2. Multi Output Format printers

In general TRIO now provides a number of new printers which may be setup initially as a screen printer. The button 'Setup' can now be used to set the type of printer output for the defined screen printer.



19. Setup of screen printer to Multi Output Output printer

The options possible for a Multi Output Format printer will be described separately for each printer type. One or several options may be selected for a printer. You should note that hardcopy of a screen is not possible on this type of printer as the hardcopy will always go to the Windows default printer.

6.2.1. Printer type

The printer type may be one of the following:

- 1 - Text**
- 2 - HTML**
- 3 - RTF**
- 4 - TXT**
- 5 - SSV**

.
The printer output definition is stored in myprt999.ini which must be present on the installation. The following is installed with TRIO:

- Type 2 - myprt002.ini**
- Type 3 - myprt003.ini**
- Type 4 - myprt004.ini**
- Type 5 - myprt005.ini**

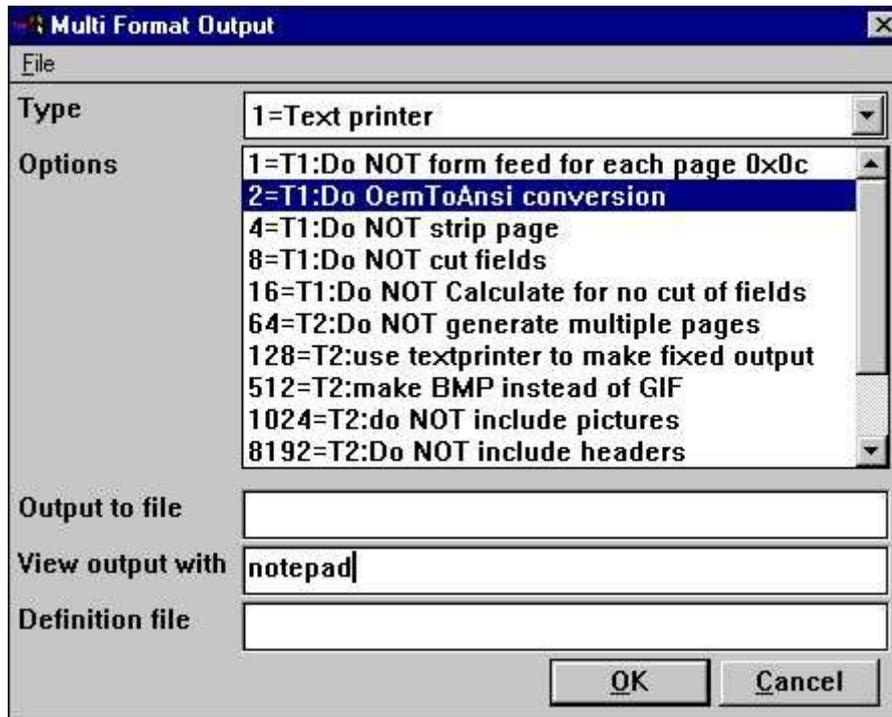
.
You may change these definition files by copying the one matching the type of printer you are using for example to

myhtml.ini

.
make amendments to it and setup the printer to use this definition file instead.

6.3. Text printer, Printertype 1

Because the Window Generic text printer has its limitations and does not produce a very good and useably content, TRIO now has its own text printer. The output will be into one file, where each page is separated by a form feed (the hexadecimal character 0x0c). The printer will automatically calculate the width of any field used in the layout in order to produce an output with no overlapping fields.



20. Sample setup of TEXT printer viewed in notepad

6.3.1. Options

The options beginning with

T1 : xxxx

is used by this printer. None of the T2 options are valid!

The printer will as standard add a form feed after each page printed. If you require to print all pages with no form feed select option 1.

TRIO will as standard convert all characters to OEM format, e.g. the format known to MS-DOS. Hereby, it is possible to use the file in MS-DOS, where all language specific characters are properly translated. The file may for example be viewed with MS-DOS EDIT. If the file is to be used in Windows the option 2 must be selected in order to get language characters correctly translated because Windows uses ANSI character set. If option 2 is set you may for example view the file in notepad, wordpad, word or others.

Internally the text printer generates a page with x columns and y lines. This page is then filled until overflow of lines is encountered. At this point, the page will normally be stripped, e.g. trailing blanks on lines and trailing blank lines are removed, which gives a small and compact file. If you want the page output with the real length of each line including all lines select option 4.

TRIO always cuts the fields width according to the field box size defined. E.g. if a text field of 30 characters is defined with a field box width only allowing approx. 20 characters only 20 characters will be printed. If required to output the entire field content disregarding the field box size select option 8. *Please note the description of option 16.*

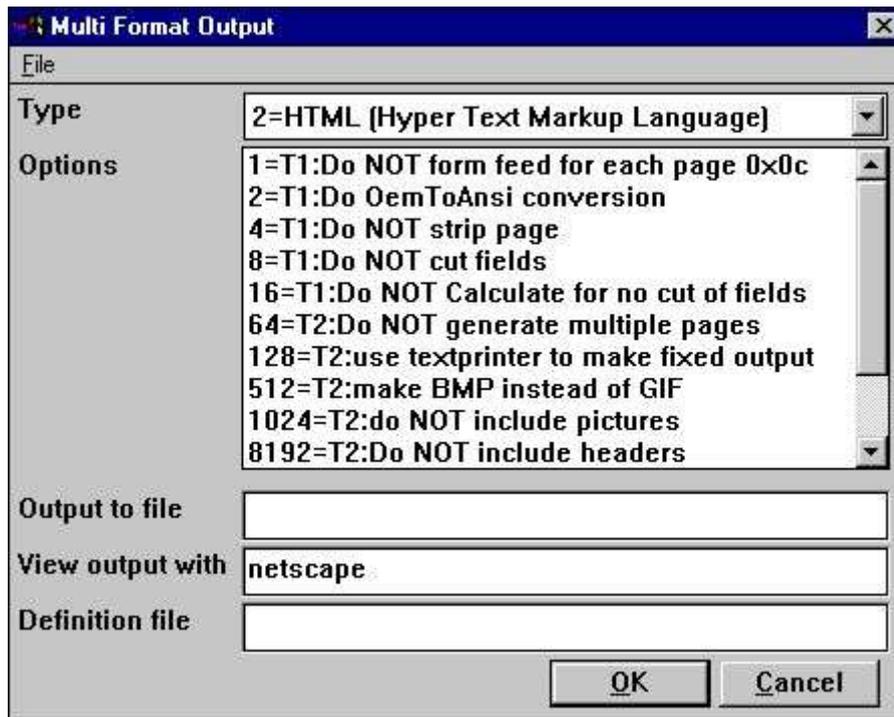
As described for option 8 all use of standard windows printer drivers in TRIO will cut fields according to a box width. However, this text printer will as standard try and calculate the box width of all used fields to avoid any missing characters from fields printed. In other words, the printer tries to expand the box width of any field used. To avoid it, select option 16.

6.4. HTML printer, Printertype 2

The HTML (Hyper Text Markup Language) printer can be used to produce output directly to internet/intranet systems. This means, an output from this printer can directly be viewed in your internet/intranet browser, e.g. Netscape, Explorer or others.

The output may also be send as attachment to an e-mail, whereas the receiver of the mail may then view the report directly in a browser.

Because HTML does NOT support any pixel positions the printer can be used in two ways.



21. Sample setup of TEXT printer viewed in Netscape

6.4.1. HTML output of standard reports

TRIO will have no problems with standard reports. The report will be converted to a HTML table, where each field corresponds to a column in the table.

Any total level or group total defined will be inserted in the same table.

All used pictures, OLE objects and charts will automatically be converted to the standard picture format GIF (Graphical Interchange Format) known to HTML.



SW-Tools	Date:	30.07.98	* Article report *	As of:	30.07.98	Page	1
----------	-------	----------	--------------------	--------	----------	------	---

Article no	Description	Selling price
0101	CHOCOLATE	2.00
0102	LARGE MACHINE	20000.00
0110	BUS	100000.00
1001	MONEY	1000.00
1005	MACHINE	2000.00
2001	CREDITCARD	20.00
2002	ID-CARD	25.00
7 Total		123047.00

22. Sample HTML output of standard report in Netscape

6.4.2. HTML output of non-standard reports

Please refer to the description of option 128 and 2.



SW-Tools TRIO Report

SW-Tools

Date: 30.07.98 * Article report * As of: 30.07.98 Page 1

Article no	Description	Selling price
0101	CHOCOLATE	2.00
0102	LARGE MACHINE	20000.00
0110	BUS	100000.00
1001	MONEY	1000.00
1005	MACHINE	2000.00
2001	CREDITCARD	20.00
2002	ID-CARD	25.00

23. Sample HTML output of non-standard report in Netscape

6.4.3. Background, logo and browse buttons

When you print a report on the defined HTML printer, the definition file myprt002.ini contains a number of references for a standard logo and browse button pictures. These picture files are found in the TRIO installation directory in GIF format:

htm-bkgn.gif - Background

htm-logo.gif - Logo

htm-frst.gif - Browse to first page button

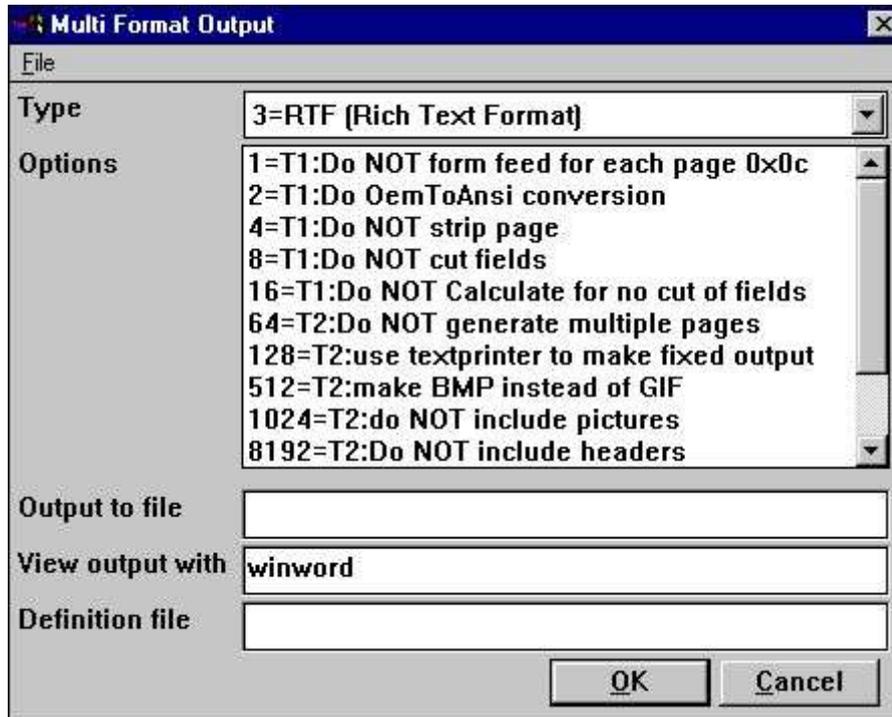
htm-next.gif - Browse to next page button

htm-prev.gif - Browse to previous page button

htm-last.gif - Browse to last page button

6.5. RTF printer, Printertype 3

The RTF (Rich Text Format) printer produces output in the same way as the HTML printer. The output from here can as an example be used to include directly in a Microsoft Word document. The only difference between the RTF and HTML printer lies in the definition file used, so please refer to the description of the HTML printer.



24. Sample setup of RTF printer viewed in Microsoft Word

Article no	Description	Selling price
0101	CHOCOLATE	2.00
0102	LARGE MACHINE	20000.00
0110	BUS	100000.00
1001	MONEY	1000.00
1005	MACHINE	2000.00
2001	CREDITCARD	20.00
2002	ID-CARD	25.00
7		123047.00
Total		

25. Sample RTF output in Microsoft Word

6.6. TXT printer, Printertype 4

The TXT printer equals the type 1 text printer, but will standard generate multiple text files, e.g. one file per page printed. This means, no form feed character is inserted in the files.

6.7. SSV printer, Printertype 5

The SSV (Semi-colon separated file) printer produces output of a standard report only (please refer to HTML output of standard reports).

Each definition part of the report is written to the SSV file where each item used in the layout corresponds to a field, and each field is separated by a semi-colon.

For example, if you have defined a report with printout of the article number, description and selling price, using the standard RAPGEN heading, the output will be:

```
Company name;Date;98.06.07;.....  
Article number;Description;Selling price  
0101;aaa;123.45  
0102;bbb;4.567.89  
2 Group 01;4.691.34  
...
```

The printer will not replace any occurrence of semi-colons in text fields.

6.8. Options for printer type 2 - 5

The options beginning with

T2 : xxxx

is used by this printer. The T1 options are only valid if type is 4 or the option 128 is selected.

The printer will generate a separate file for each page, e.g. the filename will be

xxxxx999.yyy

where 999 is the page number. If you want the output in one file only select option 64 and the filename will be generated as

xxxxxxxx.yyy

Please NOTE, if using TRIO 16-bit version the filename is limited to a maximum of 8 characters xxx plus 3 characters yyy, e.g. only 5 can be used because 3 characters is used for page number.

If you have reports defined, which cannot directly be converted into a table you might experience a wrong output from the printer. However, you may select the option 128.

When the option is selected the report output will be correct, because it is a simple text output using a fixed font.

For this fixed output no pictures, OLE objects or charts will be included.

When printing in HTML format the standard picture file format is called GIF (Graphical Interchange Format) whereas when printing in RTF format the standard is BMP (BitMaP - Standard Windows picture file format). Normally you have no need of selection this option. It will automatically generate GIF files when using the HTML printer and BMP files when using the RTF printer.

If you want to use the BMP for the HTML printer instead select option 512.

These options speak for themselves. For example, if you want to output in SSV format, but only the totals defined on a report select options 1024, 8192, 16384 and 65536. Given the sample in section 'SSV printer, Printertype 5' the only line will be:

2 Group 01;4.691.34

6.8.1. Output to file

This filename may contain path and file name for output. If working with TRIO 16-bit version the name may not exceed 8 characters plus 3 characters as extension. The syntax of the file name must be

[<drive>:][<path>]<filename>.<extension>

The default drive and path is the same directory as TRIO is installed in.

The default filename is always swxxx.yyy where xxx and yyy is set according to the options on the printer:

00 = htm

04 = rtf

32 = ssv

64 = txt

which means that with no options set the default file name is

swhtm.htm

If printing to multiple files there must be enough free characters in the file name to add the file number 999, e.g. page 5 is named

swhtm005.htm

A sample of a file name could be:

c:\webfiles\swrap.htm

6.8.2. View output with

The field is used to view the output.

Assuming the output is in HTML format and you have the Netscape browser installed in directory c:\programs, you could view it by calling

c:\programs\netscape.exe <filename>

We cannot tell you exactly which application to use for viewing the output, but only show some samples

HTML: Netscape or iexplore.exe

RTF: wordpad.exe or winword.exe

TXT: wordpad.exe or notepad.exe

Please NOTE: It may be necessary to enter the program name with complete path!

6.8.3. Definition file

The default definition file for the printer type 2 is:

myprthtm.ini for HTML output (Option 00)

myprtrtf.ini for RTF output (Option 04)

myprtssv.ini for SSV output (Option 32)

myprttxt.ini for TEXT output (Option 64)

You may define your own definition file by copying the default. If you do, enter the name of the new definition file here.

6.9. Printing Labels

The subfunction PRINT() can be used to produce print of labels in a very simple way. The function syntax is:

PRINT(LAB=direction, columns, rows, width, height, copies)

The direction control how to generate the labels on the label sheet. The valid values are:

0 - No labels

1 - From left to right

2 - From top to bottom

The multiply of number of columns and rows is the actual number of labels on one sheet.

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

Finally when printing labels you may want to print multiple copies of the same label, which can be controlled by the copies parameter.

A sample use of the PRINT(LAB=...) function could be:

PRINT(LAB=1,3,7,7cm,7cm,2)

which will produce 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.

6.9.1. How to use in RAPGEN

To produce the simple print of labels a report must be defined as a letter. The main file of the report should as an example be the customer file if you want to print customer labels.

On the layout of the letter you only require to define one label in the top left part of the layout.

When the label is defined you must remember to change the Page control, which is normally set to :1,1-99 for a letter, to 1-x where x is the number of lines used to define the label.

Finally you have to add the following calculations to the report:

FIRST

PRINT(LAB=1,3,7,7cm,7cm,2)

Please remember to have the PRINT(LAB=...) in the FIRST section. Otherwise, it will reset the label control for each record read from the main file.

6.10. RAPGEN start parameter

RAPGEN may be selected from the Windows icon with the following parameter:

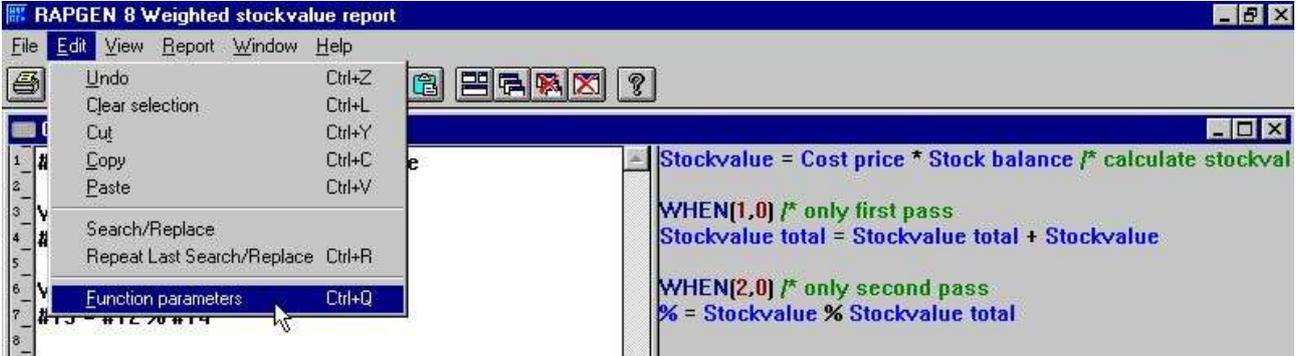
RAPWIN -m Always open dialog screen when report starts

7. Calculations

The calculations in IQ and RAPGEN has been extended with DIALOG function which also works as an extensive help function wizard:

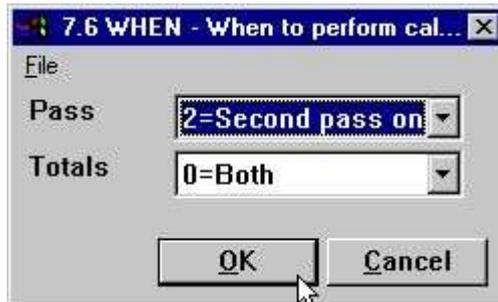
7.1. Wizard function helper

You should select a calculation line containing any function call and then from the EDIT menu activate the wizard or simply press CTRL.Q on the line.



26. Activating the function wizard

The wizard now finds the nearest function name on the present line and makes an input dialog based on the help manuals.



27. The function wizard for WHEN

Which means you will get the parameters shown in clear text and these can be selected from the allowed values.

If you move the mouse cursor over the fieldnames you will get the function manual documentation shown as floating help.

In case of a new calculation line to be entered you may just enter the function name, say READ followed by CTRL.Q

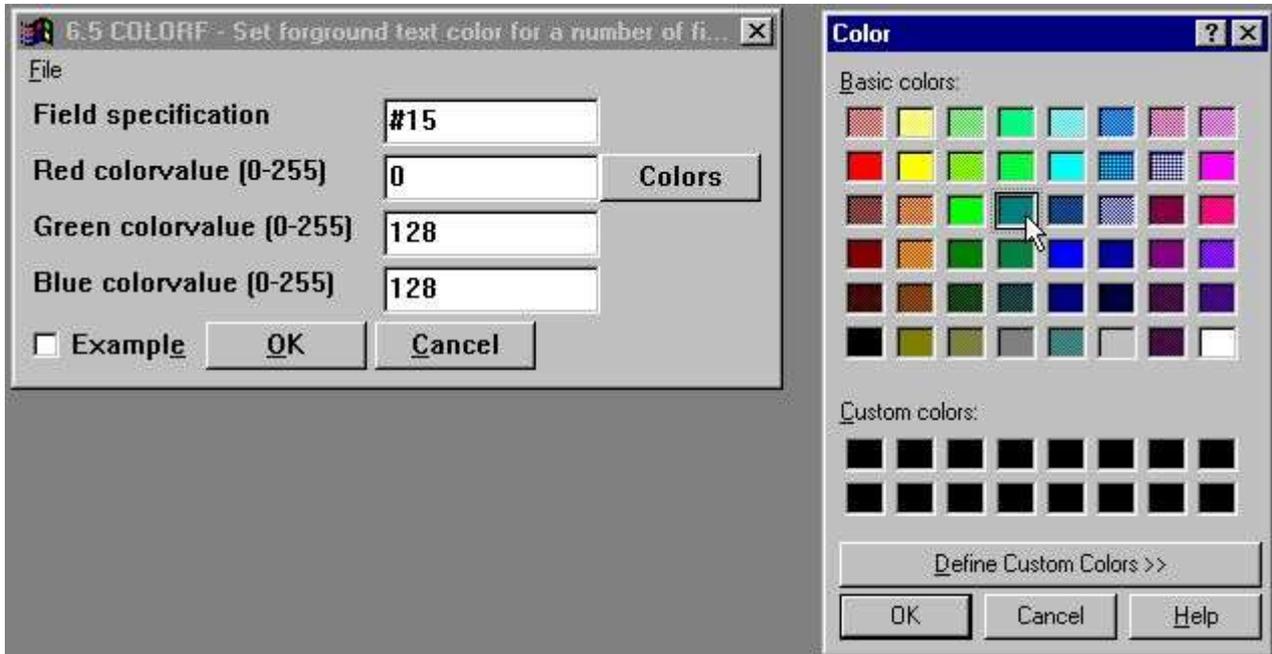


28. The function wizard for READ

The wizard also integrates to the Data Dictionary whereby it is able to let you select files and their indexes from a listbox.

If you check the EXAMPLE field the example from the manual will be included in your program, you may afterwards have to change the fieldnumbers to get a working result.

For some functions the wizard may offer you additional help by adding a button for options, BROWSE when selecting a filename or as below COLOR when selecting a color:



29. The function wizard for COLOR

You should always see the wizard as a helping tool when defining a program but at the same time observe that some functions may have so specialised use that the wizard cannot show all the possibilities. For example with READ the use of Upper/Lowercase fileID's and forced connections will not be supported by the wizard.

7.2. External DLL functions

Functions may now be integrated into TRIO by the special manual file definitions 00-99 where each field will have a reference for an external DLL, a function name, prototype and hold the function documentation also.

In this way the file description acts both as a user manual, wizard definition and programming interface. The use of these functions are just like any other function in TRIO which in fact makes an automatic INSTALL (see this) when such a function is used. The Data Dictionary will for functions hold a prototype also.

It is not intended that the user himself shall define such functions but they will be seen in extensions such as the Chart and OLE interface and opens for a wide range of programming possibilities.

7.3. **DIALOG** function for additional input

The DIALOG function enables the user to pop up dialogboxes with a selected set of fields at any point of a report execution or in an IQ program for example by click on a field.



30. DIALOG("#1,7-8,le#3") in an article query

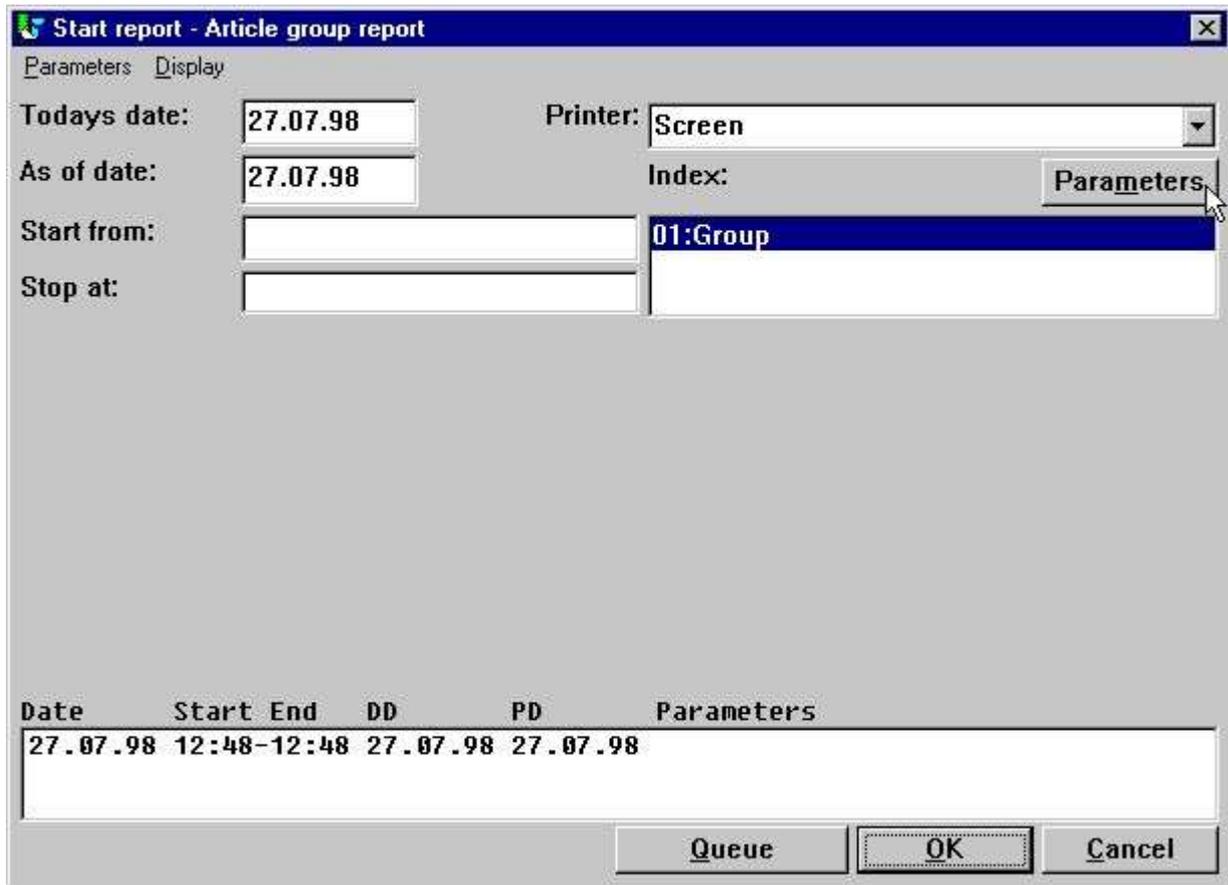
DIALOG("#1,7-8,le#3") defines a dialog with the given fields. The fields documentation is used if present as floating online help when the mouse cursor is moved over the leading text.

A number of options may be stated together with the fields such as C for Combobox, L for Listbox with valid values, W for field width and so on. These are described in the subfunction reference part of this manual. In fact are many of the new dialogs in TRIO made with the same function.

7.4. **PARAMS** function for additional report start parameters

PARAMS("#1,7C,60,le#3") is a variant of the dialog function where the input is done by start of the report not during report execution.

Use of PARAMS in a report will add a button <Extra parameters> to the startup screen which then activates the dialog.



31. PARAMS("#1,7C,60,le#3") on an article report

You may also use the DIALOG function on a report but note the difference that PARAMS takes input by report start, DIALOG when the report is actually running which means it should be placed as FIRST or under some IF control.

The image shows a Windows-style dialog box titled "Artikeldatei". It has a menu bar with "File" and a close button. The main area contains four input fields: "Article no" (empty text box), "Group no" (dropdown menu showing "0=Normal"), "Supplier no" (list box with "100=Supplier 100", "205=Sorengo", and "275=Others"), and "Address" (empty text box). At the bottom are "OK" and "Cancel" buttons.

32. The input screen for PARAMS("#1,7C,60,le#3")

8. Subfunctions

The calculations has been extended with the following functions.

8.1. **DIALOG** - Function for additional input

Number DIALOG(Fields *par1*)

Parameters: *Par1*: Fields to show in the dialog

Description: The DIALOG function enables the user to pop up dialogboxes with a selected set of fields at any point of a report execution or in an IQ program for example by click on a field. DIALOG("#1,7-8,le#3") defines a dialog with the given fields. The fields documentation is used as floating online help when the mouse cursor is moved over the leading text.

Together with a field you may state one of the following additional options:

Lxxxx	Line	(dialog units)
Pxxxx	Position	(dialog units)
Hxxxx	Height	(dialog units)
Wxxxx	Width	(dialog units)
N	No leading text	
N1	Add fieldnumber to leading text	
N2	Display leadingtext above field instead of left of field	
C	COMBOBOX, Field check definitions shown as values	
O	LISTBOX, Field check definitions shown as values	
:xx	Skip to next column fieldline xx	
+xx	Skip xx fieldlines down	

Returnvalue: OK=0, CANCEL=1

See also: [PARAMS](#)

Example:

```
DIALOG("#1-3,11")          /* Make a dialog with the given fields
```

8.2. **PARAMS** - Function for additional report start parameters (RAP)

PARAMS(Fields *par1*)

Parameters: *Par1*: Fields to show in start parameter dialog

Description: PARAMS("#1,7-8,le#3") is a variant of the dialog function where the input is done by start of the report not during report execution.

Use of PARAMS in a report will add a button <Extra parameters> to the startup screen which then activates the dialog.

Returnvalue: None.

See also: DIALOG

Example:

```
PARAMS("#1-3,11")           /* Make a dialog with the given fields
```

8.3. FTP - File Transfer Processor

number FTP(Number *par1*, Text *par2*)

Par2: FTP command

Description: The FTP function has been build in to enable advanced users to transfer files for example in a report based on a SSV file containing filenames. For a complete set of commands you should consult a FTP manual. Note that freefields may be used for the command and that the 32 bit version supports long filenames.

The example shows the transfer of a file from a Quattro system with the special command QUATTRO for transfer with headerblock and XQUAT to remove the additional FTP informations from the so transferred file.

Returnvalue: For OPEN: FTP Handle, all other: FTP error code, 0=OK

Example:

```
#10=FTP(0,"open 200.0.0.9")           /* Connect to Server
#11=FTP(#10,"user cms mypas")        /* Log in as user cms password mypas
#11=FTP(#10,"binary")                /* Switch on binary transfer
#11=FTP(#10,"quattro")               /* Switch on Quattro backup mode
#11=FTP(#10,"get /X.BASIC/0/AFIL c:/mydir/myfil") /* Get the file
if #11<>0 FTP(#10,"error")           /* Display error message
#11=FTP(#10,"xquat c:/mydir/myfil")  /* Convert from Quattro
#11=FTP(#10,"quit")                 /* Thats it
```

8.4. **GETINFO** - Get additional program information (IQ/DM)

number GETINFO(number *par1*, text *par2*)

par2 : Field reference

Description: This function allows you to get some special information from an IQ/DM program. The type 0 and 1 will return the unique id of the window, which may be used by other functions to manipulate the window. A sample of this is present in the OLE manual. When the type is 2 to 5 the function requires a field reference in *par2*. For example, to get the start column for article field number 7 *par2* should equal "va#7". The coordinates of a field is here given in the actual size of the field defined in IQ/DM. If you require the actual coordinates of a field according to the scale factor currently used, e.g. zoom in/out, use the type 6 to 9 instead.

Type 2-9 returns a field coordinate. The value can be held in a 9,T2 field format.

Example:

```
GETINFO(0)           /* Get the IQ program window id
GETINFO(2,"va#7");   /* Get the start x coordinate of va field 7
```

8.5. **LINE** - Retrieve or set the current line number (IQ/DM)

number LINE(number *par1*)

Parameters: *par1* : Type of information to get

Description: The function will retrieve or set the IQ/DM line number. The line number is the counter for lines defined in a program defined as **va#1-6l** or **le#1-6/va#1-6**.

If *par1* equals 0 the function returns the line number of the currently active line.

if *par1* equals -1 the function returns the number of lines defined for the programs. If the programs was defined as **va#1-6l,t5** the return value would be **5**

if *par1* is greater than 0 the function sets the active line to *par1*.

Returnvalue: A line number/count or zero if the functions sets the line number.

Example:

```
#20=LINE() /* Get current active line number
```

8.6. OBJECTGETSTRING- Get index of an objects selected item (IQ/DM)

text OBJECTGETSTRING(field *par1*)

Parameters: *par1* : Field on the form, e.g. va#7

Description: The function retrieves the normal value of the combobox/listbox field. It returns the value equal to the *par3* used when calling OBJECTADDSTRING.

The use of this function will normally be by click on the combo/listbox field.

Returnvalue: The normal (index) value of the current selected item.

See also: OBJECTADDSTRING

Example:

```
#20=OBJECTGETSTRING("va#6") /* Get current selected supplier number
```

8.7. 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
```

8.8. SCRPRT - Recall screen print (IQ)

SCRPRT(Filename *par1*)

Parameters: *Par1*: Filename to show using the screen printer

Description: SCRPR("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.

Returnvalue: None.

See also: PRINT

Example:

```
SCRPR("c:/w/ab.cde")      /* Show this file using the screen printer
```

8.9. SHOW- Enable/Disable/Show/Hide a field (IQ/DM)

number SHOW(field *par1*, number *par2*)

3 = Hide field

Description: This function allows you to enable/disable a field or show/hide a field.

Returnvalue: None.

Example:

```
SHOW("va#7",1)          /* Disable field va#7
```

8.10. USERINFO - Get information about user

text USERINFO(number *par1*)

17=User defined

Description: This function gets the requested user information.

The number in *Par1* refers to the fieldnumber in the system file US where you may define field 11 to 17 individually for each installation, just be careful if later upgrading the version of TRIO.

Returnvalue: String containing the user information.

Example:

```
#11=USERINFO(6) /* Get the user first remark
```

Figure list

1. The CDMENU selection	11
2. The installation selection	13
3. The FTP specification by server installation	14
4. TRIO Version number	16
5. Example of pseudoindex and recordnumber definition	28
6. Preferences	39
7. Setting up Quattro using FTP.....	57
8. Setting up XBasic using FTP	60
9. Sample setup of Navision Financials interface.....	64
10. Import of all or some of the Navision table definitions	65
11. Sample setup of Concorde interface	68
12. Sample setup of Btrieve interface	71
13. Advanced settings on Btrieve interface	72
14. The ADVANCED options with the SSV driver	84
15. The PARAMETER button for single files.....	95
16. The ADVANCED PARAMETER for the extended SSV driver defining a X/Net file.....	96
17. Additional screen printer insertion	108
18. Saving a screen print to a file	109
19. Setup of screen printer to Multi Output Output printer	111
20. Sample setup of TEXT printer viewed in notepad	113
21. Sample setup of TEXT printer viewed in Netscape	115
22. Sample HTML output of standard report in Netscape	116
23. Sample HTML output of non-standard report in Netscape	117
24. Sample setup of RTF printer viewed in Microsoft Word	119
25. Sample RTF output in Microsoft Word	119
26. Activating the function wizard.....	130
27. The function wizard for WHEN	130
28. The function wizard for READ	130
29. The function wizard for COLOR	131
30. DIALOG("#1,7-8,le#3") in an article query	133
31. PARAMS("#1,7C,6O,le#3") on an article report	134
32. The input screen for PARAMS("#1,7C,6O,le#3")	135

Index

A

ANSI..... 112

B

Basic..... 33;52

BCD..... 32

BMP..... 120

C

CANCEL..... 135

CHAIN..... 50

Combobox 131

Concorde..... 64;65;66;67;145

CRLF..... 87;90

Ctras 4

CTRAS 4;52;53;54;55;58

D

DATAMASTER..... 20;71

DLL..... 52;53;130

Duplicates 89

F

FDF 28;29

FTP..... 3;12;52;54;55;56;58;137;145

G

GIF..... 114;116;120

Global 27;59

GLOBAL..... 59

H

Height..... 135

HELP..... 14

Hexadecimal 85

HTML

106;110;113;114;115;116;117;119;120;
122;123;145

I

INDEX..... 20;28

Informix 76

INSTALL 130

Installation 9

Interface 52

IQ

.5;6;7;20;23;39;49;55;76;108;127;131;
135;138;139;140;142;143

ISAM 59;80;81

L

LET..... 39

Listbox 131

N

Navision 32;60;61;62;63;145

Netbasic 52;53

Netscape 113;114;115;122;145

NT 4;52

O

Object 6

OBJECTADDSTRING 140

ODBC

55;60;62;63;64;66;67;68;69;75;76;77;
78;79

OEM..... 112

OLE 3;6;107;114;120;130;138

OPEN 137

P

Preferences..... 37;145

PRINT 124;125;141;142

Printertype 111;113;117;118;119;120

Q

Quattro 4;52;53;54;55;56;57;137;145

R

RAPGEN 76;119;125;126;127

Recordlength 85

RTF..... 106;110;117;120;122;123;145

S

Server..... 13;40;62;137

Sort 83

SQL 75;76;77;78;79

SSV

15;25;82;84;85;86;87;88;89;90;93;94;
95;97;99;100;102;103;104;110;119;120
;123;137;145

SWODBC 22

T

TRIO

1;3;4;5;6;7;9;10;11;13;14;15;25;36;37
;46;51;52;53;54;59;60;62;64;66;68;69;
71;72;73;75;78;79;81;82;106;109;110;
111;112;114;116;120;121;130;131;144;
145

U

UNIX..... 12;58;81

W

Wizard 128

X

X-Basic 4;36;52;93