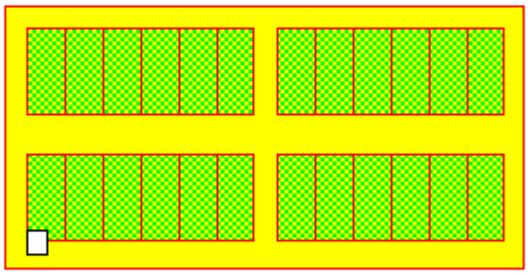
FIELD TRIAL DATABASE TLEUCH 1 ..\FDB380

File Series Trials Printouts Rawdata Outputs Base

FIELD TRIAL DATABASE



Ver. 3.80 *Copyright Torbjörn Leuchovius, SLU, Uppsala, Sweden

FIELD TRIAL DATABASE

USER MANUAL ver. 3.80

Field Research Group
SLU, PO Box 7033
S-750 07 Uppsala

Uppsala, Januari 4, 2002

Appendices

C*) Torbjörn Leuchovius, Dep. of Agricultural Engineering, SLU, PO Box 7032, S-750 07 Uppsala, Sweden

1. INSTALLATION

Welcome as user of this Field Trial Database for personal computers using Windows operating systems. This database is developed using the Superbase database handler. Install the program by running one or both of the received installation exe-files. When the database program is started for the first time, an empty database will be created for you in the language you choose (uk.sb0=English). To install the database you need a:

- personal computer using Windows any of the Windows operating system (except Windows CE)
- hard disk space of at least 10 MB plus approx. 5 MB per each 100 trials you wish to store
- good quality printer (for barcodes, some Windows True-type fonts for Code 3of9 are included)
- free serial port (only if this is required for making data transmissions from a field data recorders)
- statistical analysis package (this is not included in the database)
- spreadsheet program for making diagrams etc. from selected data from the database

Install your program this way:

The installation program is available on CD-ROM or can be downloaded over the Internet. First install the Superbase runtime system (needed for the database application) file fdb380rt.exe (from the program manager or explorer). Then install the empty database application by running also the fdb380db.exe file. Normally, the database is run as a local single-user version. If you plan to run the database shared with other users – please refer to details in *Appendix C*.

As default, the database is installed in c:\sb35 and the runtime system in the subfolder $runtime. You can change these directories in the installation program. When the installation is finished, a program group is created with the database icon in it. Double click the icon to start the database.

The first time the database program runs, it will prompt you, in English, to select your primary language. For the moment there are English, Swedish, Lithuanian and Russian versions (the Russian version works only with Russian Windows). Select the file uk.sb0 for the English version. Now the program will create an empty English database for you. This may take a few minutes.

When the empty database has been created, You can now install a simple example. To do this, select the File menu, item Import. Select all tables in the list, click OK. In the next dialog, find and select the file Example.txt in the database installation directory. Click OK to import the file.

Database settings

The program is installed with some default values. You will probably have to change a few of these. Select the File menu, item Setup. To choose from the menu you can either use the pointing device (mouse) or hold down the keyboard's left Alt key at the same time as you press and the underscored letter for the menu item. For example, the setup item is selected by pressing Alt F S.

Make any changes you wish according to the specifications in *chapter 9*. Enter your organization's name. You can enter the path and filename your company's logotype (change "logo.bmp").

You should always choose printer when starting the database for the first time. Click on the PRINTER button. Select the printer you wish to use and then exit the printer dialog. Then press the PRINT FONTS button. Select fonts no. 1-4 according to specifications in *chapter 9*. If you should use bar codes also select font 5. If you wish, you can change the name of font list (if you wish to have several font lists for different printers) before you save it and close the input form.

Windows settings

Superbase uses its own settings for number and date formats. Windows settings should not affect the database. When exchanging data with other Windows programs the settings may be important. The decimal sign should, in the English version, normally be a decimal point. When importing and exporting data, other MS-DOS or Windows programs use the point as decimal character. To change, use the SYSADM button.

Important note: The database is a 16-bit Windows program but will normally run on all newer 32-bit systems. However. *The program supports, in some cases, only 8 character long file and directory names*. It

2. GENERAL DESCRIPTION

This database consists of 16 database tables, one system table and one help text table. I addition to this there is language dependent description files containing information for each used language. In this way the database can switch between languages without affecting the stored data in the 16 tables.

Several users can start the database. However, the database is normally run as a single-user system. If one has opened table it is partly locked for other users in the database. For use of the database in networks - see technical details in *Appendix C.*

You normally enter and change data in the tables through a set of input forms. There are three table groups:

- Tables with a specific trial series as the main key identifier. This tables store information connected with the design of a series of trials like the list of treatments, profiles for result presentation etc..
- Tables with a specific location as the main key identifier. This applies to data about the used field for the trial (soil type, coordinates etc.)
- Tables using a trial identification key. This identification should be unique for all the trial's data a given year (season). Tables for treatment/plot values and "year-dependent" basic information are of this type.

The main table is found in menu Trial,General. It establishes the connection of each trial to the correct location and series. These three tables must always contain data for a new trial before that trial can be found and used by the database program. When you save new data in the main table, the program automatically adds empty records in the other two tables.

The database may receive data from a separate data collecting system of your own choice. You must supply your own handling routines for this and put the programs in the database directory. These program should handle the transmission of data to your PC as well as the conversion needed to import the data into the database tables. *Chapter 6* describes a typical routine for handling raw data. Technical details about how to call the programs can be found in *Appendix C*.

Menu system

The menu system is at the top of the program window. Select items with your pointing device (mouse) or by using a key combination. Hold down the left Alt key while you key in the two letter combination that is underlined in the menu column and row. For example, press Alt and S D to select item Design in menu column Series. Each menu column is described in *chapter 4-9*. The menu has the following 7 columns:

- File Database setup, general export-import-update of stored data, printing of tables/test files
- Series Trial design, treatment lists, variety codes, variable codes
- Trial Main table TRIAL, data for locations, basic information and treatment distributions on plots
- Printouts Printing of field cards, labels, analysis forms, diaries, observation lists. General text library
- Rawdata Handling (receiving-sending-merging-coding) and storing of computer-collected data
- Outputs Recalculation of stored data. Producing input for statistical analysis. Printing of results
- Base Data retrieval. List of stored variables, treatment averages, plot values, statistical results

Table names

Besides the titles you see at the top of the corresponding input form, the tables are named after the position in the menu system. The names have the format FDcr where c and r is the menu column and row where the table is opened. Tables which content is language-dependent also have a language code in the filename. For example, the main TRIAL table has the name FD31uk.sbf in the English language. The plot data table, FD74.sbf, only contains codes and data and the same table is used regardless of the language.

Some menu choices produce a log file (for checking the result of a process). These log files are named "cr<user>.JOB" where c and r is the menu column and row. If your computer has no special network name, you will be called "Superbase". The system will use the left 6 characters of your user name as <user>. For example, 61superb.JOB, would contain the result of the latest recalculation done on stored data. Log files can be viewed or printed from the File menu. The general log files, $$<user>.JOB and $@<user>.JOB, lists the menu choices done by different users during the current and previous work session with the database.

Function keys. Help texts. Useful key combinations

When working in an input form (see next paragraph) one can use the mouse or the keyboard to move between input fields and command buttons. Pressing Enter in a field take you to the next one. You can go forward and back by pressing **Tab** and **Shift Tab** keys.

The **button functions** is activated when you click on them with the mouse. You can also use the function keys F1-F9 or Shift + F1 to F10 (where it says ”Sh” before the key number). **Note:** The functions keys only work when the cursor is in an input field (not a button). To put the cursor in the first input field, you can use the **Escape** key. The escape key also means ”Cancel” in dialog boxes.

The **Space bar** can often be used to select something (check a box for example). If the cursor is on a button, that command is performed. When you are not in an input form the space bar halts or continues the program (for example when printing forms and data).

The program has some **on line help**. You activate it by pressing the Ctrl and F1 keys. You can choose between 7 parts corresponding to the 7 columns in the menu system. To see all the text just key in an asterisk (*). When you click OK (press Enter) you can view the text in a Window. You can now choose to print it. If not, the text window remains. You can re-size it or close it as you like.

Input forms

The database uses three types of input forms for the different tables and operations:

- Record forms show only one stored table record at the time
- Row forms show several records as rows in the form
- Dialog forms are special forms for selecting data, change settings, making printouts etc.

Input fields is used for entering data, texts, settings etc.. You can enter new, or change existing, information. You can go an input field by clicking on it with the mouse or by using the tab or arrow up/down keys. When you press Enter in a field you automatically come to the next. You can use the ”Save” button to store the new/changed information. If you forget it, it is done anyway when you activate any other button. The only exception is the close button which will ask you whether or not any made changes should be saved.

Radio button alternatives are selected by clicking on it or by moving to the button with the tab key and then using the Arrow up/down keys to change alternative. **Check boxes** can be marked on/off with the mouse or the space bar.

Command buttons performs both standard and special functions. The buttons can be activated in three ways: 1) click on it with the mouse, 2) go to it with tab/shift+tab and press the space bar, 3) press a function key. All buttons are labeled with the corresponding function key (ShF1 means Shift + F1). The special command buttons are described in the respective chapter of this manual.

Most input forms have the following standard buttons can be found (normally at the bottom of the form):

<<< << < Rewind/PageUp records, the more arrows the farther, 1 arrow means one record up
F1 F2 F3

> >> >>> Forward/PageDown records, the more arrows the farther, 1 arrows means one record down
F4 F5 F6

FIND Find (go to) the record with a key value beginning with the string you enter. If a table has
F7 several indexes the program searches in the one you are currently using

NEW This button empties the input form so that you can enter new records. For treatment values
F8 and plot values a special registration form is used. Save your data with the ”Save” button

SAVE The button immediately stores & sorts entered/changed data in the order of the table index.
F9 If you forget to save - don’t worry. Data is always saved whenever any of the buttons is used

COPY ShF1 Copy records. In record forms the current record is duplicated a new key. If the key to be copied ends with a least 2 digits, you can save a number of duplicates. When you enter the key value, append ",n" to that value where n is the number of duplicates. In row forms first position the cursor on the row you wish to copy. The copy button then duplicates all rows with with the same key value (first field from the left) to the new key. A group of rows is copied.

CUT ShF2 Removes the current record. In row forms you can remove several rows by changing the default number of rows (1). Rows not visible on the form are, however, never removed.

CLEAR ShF3 This button has effect only in row forms to remove a group of records. First position the cursor on a row you wish to remove. A group of rows is removed (as for the COPY function)

PRINT ShF4 The button prints the current form on your printer

CLOSE ShF10 This button ends the input of data and closes the input form. You will be asked if you wish to store any pending changes. This is the only way to prevent made changes from being stored.

Copy, Cut, Paste

You can always use the standard Windows functions for this. However, they do not apply to the content of an entire form. You can only mark a single text or field (drag the mouse or use Shift + Arrow keys).

To cut text:	mark the text and press backspace or Ctrl+Del keys
To change text:	mark the text to be changed and key in the new text
To copy text:	mark the text and press the Ctrl + C keys
To paste text:	position the cursor where the text should be inserted. Press the Ctrl + V keys

Dialog forms

These forms are used for different purposes like for:

- Data retrieval and for outputting the selected data
- Manually enter treatment distributions, treatment averages and plot values
- Dialogs for choosing trials and options for randomization and for printing forms, lists, results etc.
- Changing different database settings, like the choice of printer fonts

Editing templates for data presentation forms (these templates can also be edited in a text editor)

3. CREATING TEMPLATES AND LIBRARIES

In order to structure and simplify the daily work with the database you should consider in what way you can standardize data and procedures. The following work should be done and is described later in this chapter:

- Review the text library for field card, lists, labels etc. and maybe add lists of soil types and growth stages
- Review and modify the set of variable and variety codes
- Create templates in different tables for the printing of data, calculations, statistics input file etc.
- Edit the templates for printing result forms, (.RBL files)

The enclosed example contains some simple templates. The example can be imported to the database from the transport file EXAMPLE.TXT.

Below we describe how you can maintain and use different templates and libraries. Do not study the chapter in detail now. There is a lot of details that you can go back and study later when you need it.

For calculations and output of statistics files, templates are also used. Details in *chapter 7-8*.

Fetching stored information with @ functions

You can use, what we here call @ functions, to fetch database information when printing field cards, labels, lists, results etc. as wells as for statistics input files. Texts that do not contain any @ character is printed unchanged. When the fetch symbol (@) is found the program expects to find some data in the database. The following table summarizes the different @ functions and specifies where you can use them:

Function	Fetches the following data	Use for *)
@A	General information, always stored (trial, series, location etc.)	1,2a-c,3,4
@B	Basic information, optionally stored and user defined	1,2a-c,3,4
@C	Treatment field codes (like A,B,C etc.) on field map, labels etc.	1,2a-b,4
@F	Treatment factor, code for level. Also used for block number	2a-b,4
@L	Logotype location and size on field card, lists and result printout	1,2a-c,3
@M	Map (file) location and size (normally used on field card)	1,2a-c,3
@N	Numbers: Treatment or plot number, harvest number, page number	2a-b,4
@P	Parameters: Number of treatment levels in treatment factors, number of blocks	1,2a-c,3,4
@S	Specified treatment: The whole or a part of a detailed treatment specification	2a-b,4
@T	Treatment text as it is used in the treatment list	2a-b,4
@V	Information on stored variables (code, text, type, registration date, treated area)	3,4,5
@[vvv]	Print data from variable with code ‘vvv’	2a-b
name@@	Fetching data/text from a table template (only field card & observation list texts)	1,2d

***) Usage for @ functions**

1 Field card front page: Header, treatment list and distribution (map), observations, page foot
2 Labels (a), Analysis forms (b), Diaries (c), Observation lists (d)
3 Result printing template (.RBL file)
4 Statistics input file
5 Calculation formulas

Length and format of fetched data

The maximum number of characters a @ function may need to use, is given below within [ ] brackets. You can fetch a sub-string by adding a comma and specify it according to the following examples:

@A01,8 Get the first 8 characters of the Series code
@A06,-4 Get the last 4 characters of the location code
@A05,3-4 Get character 3-4 of the year-of-harvest

Note: The length specification should always come after any other amendments listed on the next page.

For the printing of numbers you may add ”>v” to print only values larger than a reference value=v. The statement @A05>1993 would print the harvest year only if it is later than 1993. For @B functions you can

also specify the number of decimals to be printed. In addition to this you can, for @B and @S functions, select only a part of the fetched data. See examples below.

General information. @A functions

The following general information can be fetched in most cases by using the codes below:

@A00 Organization name (system settings) [70]
@A01 Series identifier (code) [12]
@A02 Series title [60]
@A03 Series research leader, name+phone [60]
@A04 Trial identification key [12]
@A05 Year of harvest [4]
@A06 Location (place) identification key [12]
@A07 Official region/area/district code [4]
@A08 Other regional or station code [4]
@A09 Field station manager, name+phone [60]
@A10 Crop [24]
@A11 Trial host (farmer), name+phone [60]
@A12 Trial host (farmer), address [60]
@A13 Soil type code [15]
@A14 Soil description (text) [60]
@A15 Altitude (meters above sea) [14]
@A16 Latitude (coordinates, meters) [14]
@A17 Longitude (coordinates, meters) [14]
@A18 Trial design (code) [6]
@A19 Purpose of trial series [120]
@A20 Crop rotation number [4]
@A21 Free note #1 in main TRIAL table [120]
@A22 Free note #1 in main TRIAL table [120]
@A23 Free note #2 in main TRIAL table [120]
@A24 Free note #1 in main TRIAL table [120]
@A25 Free note #5 in main TRIAL table [120]
@A26 Statistics code #2 [120]
@A27 Part of year (char. 5-12 of A05) [8]
@A28 Statistics code #1 [8]

Basic information. @B functions

With this function you can fetch any general information from the database. However, this information has first to be defined by the user and stored in the FD33 table, menu choice Trial-Trialinfo. You may define up to 99 types of information which can then be fetched by functions @B01 .. @B99.

Table FD33 stores data under (up to 12 character long) codes like PLP.WE for ”plant protection - weeds”. These long codes take too much space to use as a reference for fetching data. You can also have any number of basic information codes while it normally not needed to specify more than 100 of these at the time. For example, to define the preceding crop (code CRP) as @B01: @B01 in the TRIAL-ID field, then type CRP in the CODE field and save the data. Now the database knows that @B01 means the “preceding crop”.

During the season you will store the basic information for each trial. It is done by typing the trial identification in the TRIAL-ID field, the type under CODE (like CRP for preceding crop) and the other text, amount, date or growth stage specification that is applicable for each type of information. A good idea is to make an empty template with the minimum basic information you always want to store. Later, when you have the information, you copy the empty template to the respective trial identification and complete it with specifications.

You can also specify @B numbers as a set of matching variables (properties). This is specially useful for printing results. If you have imported the transport file ”Example.txt”, such @B numbers was stored. For example, @B11 matching all ”FZ*” (any kind of general fertilizing) and @B12 matches all ”PLP*” (any general plant protection measure) in a trial. This information will appear on separate rows in the result printout. In other printouts only the first matching information will be printed (as for now matching pattern).

Basic information can be fetched in most cases. There are several possibilities to fetch data:

Code	Example	Fetches	
@Bnn	@B03	Defined @B03	All specifications, order: text-date-amount-growth stage [59]
@Bnn-1	@B01-1	Defined @B01	The text specification [30]
@Bnn-2	@B23-2	Defined @B23	The operation/observation date field [10]
@Bnn-3.m	@B02-3.2	Defined @B02	The amount field with 2 decimals (the .2 part) [12]
@Bnn-4	@B05-4	Defined @B05	The operation/observation growth stage field [4]

Field (treatment) codes. @C functions

When creating statistics files or printing field cards or sample labels, the treatments field codes can be fetched from table FD35 (menu Trials-Distribut.). The following alternatives may be used for statistics files and sample labels:

@C	The full field treatment code (as it is used in the field) [6]
@C*	Select field treatment codes for all factors (for statistics files)
@Cn	Field code from the n:th character (Ex. C2 fetches chars. 2-6 while C2,2 fetches chars. 2-3) [6]

The following alternative can only be used to print the treatment distribution on the front page of a field card:

@Cf1-f2,n1-n2,x,g	f1-f2=code character f1-f2, if not given the last factor is assumed
@C*,n1-n2,x,g	n1-n2=plot numbers or alternatively B1-B2 for block/replicate number B1-B2 x=column width for each code, g=number of groups for the printed codes

If you wish to place a marker sign around the selected codes, put this character after the n1-n2 interval (after the first comma sign if no n1-n2 is given). If you wish to have a marker between groups of codes, put this character after the "g" specification as above.

Example:	@C2-3,B2,4,2 means "fetch character 2-3 for block 2, print at distance 4 characters and group output in 2 groups @C* will print treatment codes for all factors from on separate rows

Treatment codes (numeric) and block numbers. @F function

For statistics files and sample labels you can print the treatment code and/or block number for a plot or treatment (sample). These numeric treatment codes are fetched from the treatment list, table FD22, menu Series-Treatmnts and from the distribution pattern stored in table FD35 (see also @C function above).

@F	The full numeric treatment (combination of factors) for a plot or treatment average [8]
@F*	The numeric treatment combination split up on each factor (for statistics files only)
@F1..@F5	The treatment code for only factor n, n=1 to 5. See explanations in *chapter 4*. [8]
@F-1..@F-5	Treatment code for last (-1) to first (-5) factor. @F-1 will always mean the sub-plot factor
@F6[-1]	Block (really: repetition/parallel number of a treatment combination) [2]
@F7[-1]	The repetition of a treatment combination within a series of trials (used for statistics files) [3]
@F8	Incomplete block number or column number for some designs (from distribution table FD35)
@F9	Group/row within incomplete blocks (as taken from table FD35)´

The difference between @F6/@F7 and @F6-1/@F7-1 is that the latter counts the number of occurrences of each treatment code while the alternatives without "-1" just determines the replicate number from the plot number compared to total number of treatments (this is all right for complete trials only).

Logotype and Map images, @L- and @M functions

The @L function is used to print your organizations logotype (the image file specified in the database settings) on different forms and lists. The @M function is used to print an image/external text file containing a map/text information of the trial (field). This file should be named in table FD32 (menu Trials, Places).

@L	Print logotype in standard size, 25*25 mm or, where condensed print is an option, 17*17 mm.
@Lh,w,i,x	Print a logotype with height h, width b (mm) and insert "i" blank rows before next printed data
@M	Print map or external text file in standard size, width 185 mm and height 64 mm
@Mh,w,i,x	Print map/text with height h, width w (mm) and insert "i" blank rows before next printed data

The last parameter, x, can be a number 1-9 for printing of alternative logotypes. These alternative files must be of the same type (having the same file extension) as the master file (the one entered under File, Settings OR the one given for a trial in the locations table (menu Trial,Places). The master filename root must not have more than 7 characters. The alternative files are named from the master file with number 1-9 added. If your standard logotype file is called std.bmp, you can have alternative files named std1.bmp up to std9.bmp. The file std5.bmp would be fetched with @L,,,5.

Numbers. @N function

The following running numbers can sometimes be fetched to forms, lists or statistics files:

@N	The current number of plot or treatment average - only for sample labels, statistics files
@N1	Harvest or batch number within a season - for batches of sample labels and analysis forms
@N2	Printed page number within a certain trial - normally for paging result printouts
@N3	Page number within the latest database printing job - use for paging diaries!
@N4	Print the date of today [10]
@N5	Print the current time [5]
@N6	Print user name (first 20 characters) [20]

Parameters. @P function

The following parameters can be used in most forms, lists as wells as in statistics files:

@P	The number of blocks in the trial (as given in the main table FD31, menu Trials-General)
@P1..5	Number of levels for treatment factor 1..5. Note: only usable for complete factorial designs
@P6	Number of treatment combinations (treatments) in the trial
@P7	Number of plots in the trial
@P8	Number of extra rows for treatment factor averages (counted from the treatment list)
@P9	Number of harvests according to main table (FD31) [2]
@P10	Number of treatment factors in the trial [1]

Specified treatments. @S function

This function can only be used if you have stored additional specifications for the treatments. This is done in table FD23, menu Series-Spec.trmt. See *chapter 4.* If you , for example, have a treatment list where you specify ”high/low” levels of treatment factors, the FD23 table can store what these high/low levels actual stands for. Another example: Your designs may say that a nitrogen level should be ”80 kg Nitrogen/hectare” but this is not always achieved in practice. The actual fertilizer level can then be stored as a specified treatment. The following information can be fetch from the FD23 table:

@S1..S5	Specification (text-amount-unit) for treatment factor 1..5 (FAC in table FD23) [44]
@Sn-1	Specification for factor n, only text part [20]
@Sn-2	Specification for factor n, the amount part [12]
@Sn-3	Specification for factor n, the unit part [10]

Treatment texts. @T function

You can fetch texts from the treatment list (table FD22, menu Series-Treatmnt) for field cards, labels and statistics input files. Treatment texts are automatically included when printing results.

The following alternatives can be used for lists and statistics files:

@T	Fetch the complete treatment text [72]
@T*	Fetch treatment texts for all factors (for statistics files)
@T1..T5	Fetch the treatment text for a factor, 1.. 5 [72]

For the treatment list on the field card you can load the treatments differently (a combination @C+@T):

@Tn[,c,r,s]	Fetches field (treatment codes) as wells as treatment text for each level of factor n
@T*[,c,r,s]	Fetches field treatment code and treatment text for all levels in all factors

With the optional parameters [,c,r,s] you can specify that the texts should be printed in “c” columns and use at least “r” rows. The “s” parameter should be 1 if the treatment codes should be sorted. This is only interesting if you have randomized field treatment codes with the “ABC-order in first block” option. With that option the treatments comes out as they was placed in the first replicate (if not sorted).

Example: @T3,2,7 means: print treatment levels for factor 3 in 2 columns. If this produces less than 7 rows, extra blank rows will be inserted in the printed list

Variable's information. @V function

You can fetch information on stored plot/treatment variables to statistics files and when printing results. This data is stored in table FD73, menu Base-Varlist. The fetched information refers to the current variable column in results forms but to the nearest variable column to the left in statistics files. You can fetch the following:

@V	The full variable's information, code-text-date-type-observed are [87]
@V1	Variable code [12]
@V2	Explaining text [40]
@V3	Origin of variable, + is observed, C means calculated [1]
@V4	Type of data, P is plot data, T is treatment averages [1]
@V5	Date of registration [10]
@V6	Observed area, m^2, for example net harvested area for the yield [11]
@V7	Decimal growth stage at registration time [4]

Printing of data in a variable, @[…] function

If you wish to print data from a variable (tretament averags, plot value or text item, table FD74..FD76) on *sample labels*, you can fetch the data with this function.

If, for example, straw strength is stored under the variable code STS, data for each plot/treatment can be fetched with @[STS].

Texts from templates. @@ function

When printing field cards, and also observation lists, text can be fetched from prepared templates (libraries with standard texts). The @@ function works in the general text library table FD46 (menu item Printouts, Printlib.) for making references to rows in templates (text libraries). The field cards often contains a lot of texts for different observations that should be carried out in a trial. It can then be practical to arrange these texts in templates, for example one for each type of crop.

Texts for field cards or observation lists have the series code or the location code as key. To avoid listing a lot of text rows, you can refer to rows in a standard list (text template) by fetching texts as follows: Let's say that you have a variety testing trial series P-123-1997 and that you have a text template (or library) for cereals call GRAIN. On a row of texts for that series you can refer to rows from GRAIN: You write GRAIN@@r,r,r,r,r where r is row numbers for information rows to print from the GRAIN template. You can have several reference lines with references to this or other templates. You can also freely mix rows with or without references.

When referencing observation rows, a P for plot or X for treatment average can be appended to any row number. The text of that row will then be printed with the text "PLOT VALUES" or "MEAN VALUES" at the end of the row. Example: row no. 15 has the text "Emergence". If the row is fetched with 15P, the text "Emergence PLOT VALUES" will be printed.

Used printer fonts. ^ function

The printer fonts that the database program will use should be set in the database File-Setup menu. There you will find button "PRINT FONTS". This button takes you to a form where you can select 5 different fonts, numbered ^1..^5. When printing forms and lists the following printer fonts will be used as default:

Printout	Font
Text files, from menu File-Print	Standard font (^1)
Field card, labels, analysis forms and observation lists	Fixed 12 character/inch font (^3)
Sample batch labels, Preface for diaries	Larger font, may be proportional (^2)
Diaries and result forms (reports)	Fixed 12 or 16.5 character/inch fonts (^3 or ^4)

On field cards, labels, analysis forms, diaries and on result report templates, you can changed the normal (default) font with ^ functions below. Those functions can also change style and set printer tabs. If you plan to use bar codes, you must always use the ^5 function in the text rows to print some text as bar codes (^5). You also need a printer with bar code cassette fonts or Windows soft fonts.

^ function	Meaning
^T	Sets a tab in the ^ position on the text row (you may need it for proportional fonts)
^B	Boldface text
^I	Italic text
^U	Underlined text
^N	Reset to normal style
^1..^5	Select font number 1..5 as in the font setup list
^0	Reset to the normal fint number for the specific printout (se above)

General text library, menu Printouts-Printlib.

This is the text library where all texts for field cards, labels, analysis forms, diaries, job lists etc. are stored. You only have to bother about these text if you wish to use the database to produce these types of printouts. The fields in the input form has the following meaning.

KEY(GROUP). If you, for the field card, use a key identical to either the series or location identification of a trial the matching texts will automatically be printed for each trial. It is , however, just as common to have a set of templates for different types of trials (treatments, treatment distribution on plots) or crops (the set of observations that should be carried out). In the latter case you should, at printing time, select the templates as well as the appropriate trials to be printed with each template. For other lists than the field card, you also select a template with a key (free of choice).

ROW. This column determines the order the text will be printed: Use number 0 for the titles of the templates.

POS. This is a very important field. It determines the type of text - for which printout it is intended. It can include, but is not limited to, the following options:

- **-** Field cards, a header at the top outside the frame
- **1** Field cards, 1st field inside frame, for treatment descriptions
- **2** Field cards, 2nd field, for map and treatment distribution on plots
- **3** Field cards, general observations (lower left part)
- **4** Field cards, plot/treatment registrations (lower right part)
- **5** Field cards, page foot
- **6** Labels, for individual samples and batches
- **7** Analysis forms, header and column texts
- **8** Diaries, for keeping track of samples and trials
- **C** Lists of crop codes
- **F** List of fertilizers (numeric codes and weight percentages of nutrients)
- **G** Lists of growth stages for different crops (decimal BBCH scales)
- **H** List of humus characterizing codes
- **R** Arrays for randomizing certain incomplete designs
- **S** Soil types, codes/explanations

TEXT. Here you enter the text you wish to print. Normally it comes out the way you enter it, but there are a number of @ functions and ^ functions (see earlier in this chapter) for fetching data from the database or for changing printing style. Below follows some more details for each type of printout.

C - Crop specifications: You can here list short codes (1-3 characters) for common grown crops. These codes can be abbreviations of the English names of the crops. These codes will be shorter when used in variable names (see further down in this chapter). A good alternative is to use the EPPO/Bayer Latin codes. These are longer (5 characters) but often only the first 3 (family) may sometimes be used.

F - Fertilizer specifications: You can list common fertilizers with a numeric code under ROW and its corresponding text description. From position 45 and right in the text field you can enter the weight percentages of different nutrient elements. If a fertilizer contains 20% nitrogen and 6.5% phosphorus you can input N:20 P:6.5. The use of such a list is as follows. In the Trial Info table (FD33) you can enter fertilizers with the short numeric code and date/amount. The variable code should begin with FZ or MRE (manure). When running the “Decode fertilizers _ soils” option in the menu “Trials, Other”, the database program will calculate and add total applied amounts of different nutrients in each trial. These figures may be needed when analyzing and/or printing results.

G - Growth stage specifications: In the European cooperation in agricultural research, so called decimal scales has been developed. These are very useful in databases since a certain number (between 0 and 100) always means the same growth stage independent of the type of crop.

H - Humus content specifications: In section, different soil classes (as regards the humus content) can be listed. In the text field you can enter an abbreviations and a longer text within parenthesis. From position 45 and right you can specify the corresponding humus content interval, like 3.01-6.00 (%).

R - Randomization arrays for incomplete designs: When randomizing trials you often have complete designs. However, certain incomplete designs may be very useful for trials with many plots or treatments. These designs can be randomized by the database program only if special patterns (arrays) is stored here. The incomplete designs that the program can handle is LC (Lattice Alpha), RC (Row-Column) and RL (Resolvable Latinized) 1-factorial designs. In the example.txt file that comes with the program, arrays has been prepared for certain ranges of treatments and replicates.

All arrays start with type and row/column parameters in the form TTrrcc[*b]: where TT is LC/RC/RL, rr is No. of rows in each block (from 01-99) and cc is No. of columns (01-99). For the RL type also the number of blocks must given (*3 for three block = complete replicates). For LC designs the number of blocks is not required and for RC it is the same as the number of columns. After the parameters the pattern numbers (not randomized treatment numbers) comes, divided by commas, from plot 1 and up. When randomizing, the program takes the arrays that fits the best or comes first (if more than one array is useful, for example RL0307 and RL0703 which can both be used for 21 treatments but with 3 or 7 incomplete smaller blocks).

S - Soil specifications: As for fertilizers, soil types can be listed as numerical codes with a text description. The same utility as for fertilizers is used to complete entries in the Trial Info table. In that table, you may enter the variable HUM and give the humus content in the amount field and soil type code in the text field. From the S (and H) lists in the text library, the program will translate the codes but also change the soil type description in the trial's location entry in table FD32.

The push buttons at the top of the form is used in the following way:

EDIT-LST Sh F5. With this button you can edit field cards (in a way more like the printed cards), labels and lists. Normally you begin with copying (with COPY button) an existing template. The click on a row in this template, click EDIT-LST and select the template from the list. Depending on the type of list (determined by the POS numbers - see below) the corresponding edit form will appear. You can move around texts in the different fields, move between fields with Tab/Shift Tab (or click) and go back to upper corner by pressing Escape. Press OK button in the upper left corner to save the changes.

TREATMNT Sh F6. With this button you can copy treatment texts from a treatment list. This will give you a template to start with when there is no existing one that can be used. Texts can be copied as they are (check the box for that) or as reference (with the @T and @C functions)

VAR.LIST Sh F7. This button is mostly used for updating general templates for observations (like the |UK3| and |UK4| imported from the example file "example.txt"). The button shows the general variable list and a variable code associated with a text row can be selected.

OBSERVAT Sh F8. The button is used to select and to create references to text rows in general templates. You should first click on a row with POS=3 or 4. Then mark the texts you want (the texts in the template should have been arranged in a suitable order). When clicking OK the @@ references to the general template will be added.

GROWTH ? Sh F9. This button works as the Sh F7 button but shows growth stages instead. They can be added in a general template to specify a growth stage (or interval) when this measure/observation should be carried out (see below). If growth stage or date intervals has been specified it is possible to make printouts of measures to be carried out at a certain time/growth stage interval. Such printouts are made from menu Printouts, Joblist.

Field card (POS contains character -, 1, 2, 3, 4 or 5)

When printing field cards, the text field can also contain references to text rows in other templates by using the @@ function. First on a such a referencing row you enter the exact name of the referenced template, immediately followed by @@ and the followed by row numbers separated by comma signs. Examples of this technique can be found in the transport file ”Example.txt” which can be imported to the database (under the File menu, see also the *chapter 1*). If we, for example, have a template called MYTEXTS and reference to it by writing MYTEXTS@@1,2,3,4,,6,,8,9,10 the effect will be that the texts on the given rows in MYTEXTS will be printed. The double comma signs in the list insert a blank row in the output.

POS 1, treatments. The treatments can be entered as they are. You can also take a copy from the treatment list with the TREATMNT button and make some additional adjustments. A convenient way is to fetch the treatments with the @T function as described earlier in this chapter.

POS 2, map/treatment distribution on plots. You can, of course, enter the treatment other in plain text but, since you often have randomized patterns for each trials, it is often more convenient to use the @C function to fetch each trial’s distribution from the database. You can specify which factor and which plots or blocks you wish to print and you may also group them. If you put a marker just before the @C, that character will be printed around each group/block.

POS 3 & 4, trial observations. While the texts for POS -, 1, 2 and 5 can be 88 characters long, you can only print 44 characters in the 3 and 4 fields. The rest of the TEXT field can preferably be used to specify the appropriate time interval, variable code and/or a so called work code. These specifications must be entered from text column 45 (or later) in the TEXT field and with comma signs between (the comma should be there if you skip over time interval or variable code). Time intervals are entered with D (date) or G (growth stage) optionally followed by an interval. Dates should be given as month-day in the form mmdd and growth stages should be of the BBCH decimal types. Variable codes can follow the rules described later in this chapter. A work code can be a character indicating what type of observation it is like M for measurement, W for weighing etc.. A ruler in the input form shows where you can start enter specifications

If any of these specifications has been given the texts can be printed in a variable list (*chapter 5*, field card, b) option). If time interval was entered, that observation can be listed in so called work lists (see *chapter 5)*.
Examples: D0610-,WE.GA,W after June 10, gram weeds/m2 (WE.GA), weighing (W)
G72-80,STS.G75,G between BBCH 72 & 80, estimate straw strength (STS.G75), grading(G)

If you fetch texts with the @@ function (you can use OBSERVATIONS button to pick row numbers from a template) you can also append X or P to a referenced row number. This will append "Averages" or "Plot Obs." to that text row.
Example: CEREALS@@1,5,10,11X,12,20P will pick text rows 1,5,10,11,12 and 20 from template CEREALS and add ”Averages” to the text on row #11 and ”Plot obs.” to the text on row #20.

Labels, analysis forms and diaries (POS contains 6, 7 or 8)

For sample labels and analysis forms you have the option to include a number between 1 and 999 in the name of the text template. If the template key is LABEL-118 then 118 can be checked against each trials’ treatment list so that only matching treatments will be printed. For further details please refer to *chapter 4* and the LS field description.

Labels, POS=6. Texts for individual plot or treatment samples should go on rows with positive numbers. texts for containers/batches are specified on negative row numbers. The label texts can contain @ and ^functions. Labels are printed on A4 format with 2*8 or 3*8 labels. These label sizes allow only 4-6 text rows.

Analysis forms, POS=7. Texts for a list header should have negative row numbers. The list header can contain @ functions. On positive row numbers you enter texts for data columns. These columns are always 8 character wide, so the text must be adjusted to look nice in groups of 8 characters in the TEXT field.

Diaries, POS=8. Diaries are used to keep track of samples or trials, for example the current status of what has been documented or what information has been received for each running trial. You can optionally design a diary to have a cover page. Texts for this should go on negative row numbers. In the diary, information from each trial comes on one or more rows. The text or data you like to print is entered in the text field on positive row numbers. You can use a row for each piece of information. If you wish to begin a new printing row, then add a table row with only a slash "/" in the text field and then add table rows with texts for that new printing row.

You normally use @ function to get information from the database. You enter each information piece by giving a column header followed by a equal sign and then the information to be printed for each trial. **Note:** The number of characters up to the equal sign determines the width of that data column. For example, to print the column header "Crop" and fetch the trial's crop information enter in text field: Crop =@A10

Variety/text library, menu Series-Num.list

In this library, you can keep an historical file of codes for all tested varieties, pesticide compounds etc.. The library can also contain sets of standard texts for specifying treatments in general. When preparing a treatment list for a series of trials, you can pick texts and codes from this library.

Different kinds of codes or texts should be grouped in different categories (the CAT field). For general texts the NUMBER field just sort this texts in number order. For varieties etc. you should store their official/normal number codes and enter the texts according to your national standard.

The following variety text coding standard may be used where the breeders usually have 1-3 letter codes: For new varieties with only a number you can store the breeders code and number like "HI 5058". For varieties with official names you can store name.breeder like "GOLF.NI".

To this you can add the company (if other than the breeder) that markets the variety in a country within (). The "GOLF.NI (SW)". In Sweden, the barley variety GOLF is marketed by SW (Svalöf Weibull). You may also add other properties like using 4N for tetraploid varieties.

Templates for printing results, menu Outputs-Profiles

For printing results you can store profiles (column descriptions) for the data columns. This is done in table FD65. Profiles can also be used to specify columns for input files for statistical analyzing programs.

Besides this profiles, you can also (to some extent) design the layout of the printed form. Some standard layouts (for condensed print or A4-landscape print) are included. These blank result forms always have the file extension RBL. With the EDIT MASTERS button you can open a form to edit these files. Press OPEN to load a form. You can modify it. If you change the file name you can save it as a new file.

Most of the text you enter come out as you write them. You need, however, to use some @ functions to load general information from the database. Most often you use the @L, @A, @B and @N2 functions. You can also change text font and style with the ^ functions. Please see the beginning of this chapter.

You can also edit the result printing templates in an ordinary text editor. It is then important that the file:

1 has exactly 16 text rows for the upper part of the printing form (some rows may be empty)
2 row 17 begins with the text @V and nothing more
3 row 18-22 contains texts that should appear to the left of the data column header
4 row 23 begins with the text @D and nothing more
5 has a maximum of 5 extra rows (row 24-28) for page foot text for the printed results. These texts are only printed on the first page of a result printout.

Variable and basic information codes, menu Series-Var.codes

This library should contain a list of all codes that are used for identifying all stored as plot values, treatment averages or as general information for the whole trial. When new types of observations come into use, it is good to include them in this library. The recommendations below can be helpful when setting a coding standard for your database. A standard makes it possible to exchange data with other researchers.

The library can contain only code elements that can be put together to describe a variable. The fill codes can then be built up with one element as a main code and one or more elements as sub-codes. The total code can not be longer than 12 characters.

For common registrations and culture crops, English abbreviations, 1-3 characters long, is recommended. For specific pests, diseases or weeds, the EPPO/Bayer system of Latin codes should be used - always as main codes. These codes are 5 or 6 characters long.

The sub-codes should specify materials, time/growth stages, registration unit etc.. Allowed characters are capital letters A-Z and digits 0-9. Decimal points and hyphens (-) are used with special meanings.

To specify a season (for example harvests of forage) or to count rotation years, directly add 1 digit onto the main code. If Y is "yield" and DM is "dry matter content", Y3 means "yield, harvest 3" and DM2 "dry matter content harvest 2".

Specifications of material or crops should follow directly after the main code. Add the hyphen and a 2-3 character long code. Example: Y3-GR means Y=yield, 3=3rd harvest, -GR=grass part

Other sub-codes can be added to specify, in the following order: growth stage, class interval, method of registration, registration units (if not the default). Each sub-code starts with a decimal point.
Example: .KG means "weight, Kilograms". Combined codes are just put together (.KGM2 for Kg/m2).

Extended BBCH decimal scales should preferably be used. Sub-code G could be used in general (like .G50 for stage 50) and sub-code R especially for ripeness. The growth stage is often important, for example, to relate damage levels to certain leafs (when leafs are counted from the top at each growth stage).

Sub-codes may include class intervals. If not given, "total" or "true/false" data should be stored - whichever is appropriate for the variable. Intervals normally reads "less or equal to". Complete intervals will be of the type I55-75 ">55 and <= 75".

Preferably use SI units or multiples of them. Units should be chosen so that the stored data in all expected cases falls within the range of 10^{-1} and 10^{6} (with the exception of a zero value, of course).

Assessments are normally made as percentage of the observed area (% or 0-100 values). Alternatively, sub-codes can denote other types of assessments.

Components in materials are normally given as % of the dry matter weight of the material. You may also specify weight-% of raw material, % of the number in a sample or volume shares.

Dates as plot data or averages, can best be stored as MMDD.YYYY. It is allowed to exclude the year when this is not significant. Examples: 805 is "August 5" and 1115.1994 is "November 15, 1994".

Yields should be stored as clean, dry matter material in the unit Kg/hectare. Recalculation to other moisture contents or weight units is preferably done after extraction of data. A sub-code, like M, can be used to refer to moisture reference levels (Ex. M15 for "15% moisture"). Unclean field figures is also denoted with a sub-code like .RD ("raw data")

Time (duration) could be stored as days, hours or seconds. 1/100 of the units should be used when decimals is required. Example: Store 1.50 for "1 hour and 30 minutes" or 2.25 for "2 days and 6 hours".

Nutrient levels normally could be stored as weight-% of a material. The sub-code .KGHA change this to Kg/hectare. For fertilized amounts the sub-code .FZ could be used and the default unit here is Kg/hectare.

Currency figures should, if possible, be coded with the international 3 character codes. Use 1/100 of the unit when decimals are needed. For currencies where 500 units is less than 1 US dollar - skip 3 zeroes. A USD value of 1.60 means 1 US $ and 60 cents and a 25500 ITL (Italian Lire) is stored as 25.5.

4. STORING NEW TRIALS AND TREATMENT DESIGNS

When plan your trials for the next year, it is easiest to follow the steps in this chapter. Sometimes you lack some information and must do the steps in a different order.

Storing new trials in the main table, menu Trials-General

This is the main table in the database. Here each individual trial gets a unique identification, the TRIAL-ID. Each trial should here be given a SERIES code and a LOCATION (place) code. To be able to store trial data from different periods (years), a long-term trial at a special location should get a new TRIAL-ID for each period (year of harvest). When you save the new record in the main table (with the SAVE button), the database also creates empty records in the series and location tables (if those records don't already exist).

The number of blocks (replicates) may vary for different trials in the same series (without affecting the series design or the statistical analyze to be done). It should therefore be entered in this table. The "number of harvests" can be used for trials with many harvests per year and then be printed as a parameter on sample labels or analysis forms.

Also enter the type of crop (for rotation or cultivating system studies, the crop may not be important, for forage trials you might add the ley year number after the planting year). The district fields can be used to enter official (district-1) or other regional codes for your country. YEAR means the year of harvest. After the year (written with 4 digits) you can enter additional information. This information can be useful to search on or group trials when making printouts (labels etc.) or for printing selected results. Letters F or S could mean trials planted in the "Fall" or "Spring". You can group something else, like a crop code.

Storing new series designs, menu Series-Design

If you (in the previous step) have stored trials with a new design, you should now go to this table and look up the created empty records (use the FIND button or step with the < or > buttons) and complete the information about these new designs (trial series).

Important note: You must store a new design as soon as the number of treatment combinations or some treatments are varied in any respect. It is recommended to denote design (series) with codes composed by originator-running_number-version_number (like DEPX-123-2). The version number can be used when you have a set of designs that are very alike (DEPX-123) but some variants exists (-2 in this example)

Now start with the DESIGN button to select the type of trial. If something from the lists is correct, then select it in order to store some default values. You can also enter another code of your choice. The program can use some of the listed designs for randomizing treatment distributions (see next section). In other cases, you must manually enter the distribution of treatments.

After the design is chosen, you must correct the suggested number of levels for treatment factor 1..5 (F1..F5). Factors not used should have a zero value. **Note:** In split-plot like designs factors are numbered from larger to smaller plots with the same treatment factor level. F1 is typically a main plot (block) treatment. If ,for example, you have a series with a main plot treatment in 2 levels and 6 sub plot varieties you should enter 2 in F1 and 6 in F2 (F3..F5 should be zero).

All treatments (combination of factors) used in the trial will be identified by a 8-digit code (see treatment list). The TREATMENT MASK is used for decoding these codes. The normal mask is 1115. Each mask number **from left to right** specifies how many digits is used for each treatment factor in the treatment codes. The mask 1115 with 4 numbers allows maximum 4 factors. The last factor (the highest of F1..F5 which is not zero) may use 5 digits but the factors above (main plot or block treatments) only 1 digit each. This means that lowest factor (ex. varieties) could include variety numbers up to 99999, the others are limited to 1-9.

The PRINT MASK is used for printing treatment combinations (for example on a field card) in text codes. These codes are generated when randomizing the treatment on the plots. 1A means that we have 2 factors (the mask has two characters). F1 levels will be printed as 1,2,3 etc. and F2 levels as A,B,C etc.. If you have more than 26 levels in a factor you can use @ for generating A-Z and a-z for levels 1-26 and 27-52. You can also use # which will print levels as 1-3 digit numbers (like from 01 up to 35 for 35 levels in a factor).

The MASTER field may contain the name of a text file that specifies how to print the results from this trial series. The default is STANDARD.RBL. How to create and edit these type of files was described in the end of *chapter 3*.

STUDY PM (FILE): You can refer to a Windows rtf file containing the full project plan for this trial series. The file can be printed (using MS Wordpad) in combination with printing of a Field Card with summarized instructions for the trial.

MISSING DATA ADJUSTMENT affects the way treatment averages is calculated before presentation of results. Which method that will be best depends on the statistics design. The 4 methods means:

1. Difference method. This technique should produce a good result when there is only one missing value in the trial. It is often referred to in Statistics books (maybe called something else).
   $X_{ij}= (aT_{i.}+bT_{.j} -T_{..})/(a-1)/(b-1)$
   X_{ij}= calculated replacement value, a=# of replicates, b=# of treatment levels in each replication,
   $T_{i.}=\Sigma$ in replicate with missing treatment, $T_{.j}=\Sigma$ for treatment with missing value, $T_{..}=\Sigma$ all observations
2. Quote method. A missing value is adjusted to the ”strength” of each replicate. During the calculation the replicate strength is calculated only on the basis of those treatments that are complete in all replicates.
   $X_{ij}= T_{.j}*T_{i.}/T_{k.}$
   X_{ij}= calculated replacement value, $T_{i.}=\Sigma$ in replicate with missing treatment,
   $T_{i.}=\Sigma$ in replicate with missing treatment, $T_{k.}=\Sigma$ for replicates where the treatment has an observation
3. The average of method 1 and 2 above
4. Method 1 for trials with only 1 missing value, otherwise method 2

If the program should fail to find a good replacement value there is also another possibility when you print results. You can put the results in a text file, manually edit that file and finally print the results from that text file. For details, see *chapter 7.*

Information text or fetched (@) parameters in the STATISTIC MODEL/CODES fields, may be passed over to a statistics input file. See details in *chapter 7* and *Appendix B*.

Specify treatments, menu Series, Spec.trmt

This table, FD23, is used to define all levels of each treatment factor. The table is also used for detailed specification of the treatments after they have been applied. Furthermore, this table can store text rows for presenting results from statistical analysis. Now, we will concentrate on the first use of the table – to define treatment factor levels for trial series.

First look for an empty list with the series identification to the left (you can find it with the F7 button). For each factor you should specify the levels under the text column to the right. Let’s assume that you have a split-plot two-factorial design with 2 block treatment levels (factor F1) and 8 varieties (factor F2) in smaller plots. Then start with 8 levels of the last factor – the rows where TRM/FAC is –1 and 1…8. Enter the texts for the 8 varieties. You may also enter numerical (here: variety) codes in the VALUE/LEVEL column. Then you enter texts for the 2 block treatment levels on following rows, where TRM/FAC is marked –2 and 1…2.

When texts have been entered, push button MAKE TREATMTS. Now the program will combine the factor levels to a complete list of treatments with actor combinations and the separate factor levels. This list will have 2*8 + 8 + 2 = 26 rows and is stored in the treatment list table, FD22. When prompted to create this list, a checkbox will affect the order of the listed factor averages after the 2*8 combined factors – whether varieties regardless of block treatment should be printed before the block treatment levels or vice versa. In table FD22 further adjustments can always be made later on.

There is one more button in the form – GET TEXT/CODE. You can use this to fetch standard texts from table FD24 (like variety codes and texts). That table can contain all sorts of texts and should be maintained as new standard texts (like varieties, fertilizers etc.) are introduced. The table is found under menu Series,Num.list.

Before pushing the GET TEXT/CODE button – click on the row where texts should be pasted. You can get the texts in two alternative ways. The first and simplest way is to mark the texts you wish to include and click Ok. Texts will be pasted as long as there are remaining levels of the factor to fill out. However, the text will be pasted in the order of the list you selected them from.

Another alternative is to enter category and numerical codes in the TEXT column for the first factor level. Suppose you have a category of texts called WHW (winter wheat varieties). You wish to enter codes and texts for varieties 12,9903,865,17,35,9652,7632 and 1865 *in this order*. You enter this in the following way:

WHW@@12,9903,865,17,35,9652,7632,1865

When you now press GET TEXT/CODE the varieties you have listed will be displayed. If these are correct you mark/select them and click Ok to paste them. The row will codes will also be save and can be used again.

There is one more important remark about the numerical codes in table FD24. You normally enter positive number/codes when you wish to include both the code and the text when specifying treatment factor levels. When the code is not interesting, you can instead use negative numbers in table FD24. Such texts will be fetched but the codes will not be pasted.

Check/adjust the treatment list, menu Series-Treatmnts

Each trial series should have a treatment list that is stored in this table. The list should have been created through the steps above. You can adjust the complete list at any time, but the treatment numbers in the N:O column are important. They correspond to treatment averages data with the same numbers. Further, the distribution of treatments on plots, table FD35, use the same numbers. Therefore you can change these numbers only at design time. If done later on, you will have to re-randomize (or change the treatment distribution) and change any entered treatment averages data for all affected trials in this trial series.

SERIES: This column contains the trial series identifier, the same as used in series-design table - table FD21. As soon as the number or meaning of any factors or levels factors needs to be changed, a new variant of the series identifier has to be used. If not, the database cannot be consistent when data are calculated, displayed or retrieved.

N:O. These numbers were given each treatment when the complete list of treatments was created (earlier step). If you have incomplete factors in multi-factorial designs, you may exclude some of the treatment rows from this list *but you should not renumber the list*. Just leave gaps in the numbers (se also 'TP LS' below).

CODE #: Numerical codes for each combination of treatment factors according to a **code mask**. The codes can have up to 8 digits. The code mask, set in table FD21, tells the database how to interpret these numerical codes. The sum of each digit in the code mask should be 8. The default mask is 1115. It means that the last 5 digits in the treatment codes identifies the levels at *last* factor (factor on smallest plots in split-plot designs). Other factors (often block treatment factors) can each use 1 digit, which means that they can have no more than 9 levels (1-9) each. In a trial with 2 factors the codes will start by 100001, 100002 etc..

TEXTCODE: This is the code that will be used in the field/laboratory a short identifier of each treatment combination. Please also refer to 'PRINT MASK' under Series, Design above.

TP LS: Treatment details (type, printing): This columns should contain a plus sign for all treatment combinations that were actually laid out in the field. For factor averages, this column contain an 'F'. Rows with a '@' in this column will be included when printing results from trials, others will not be printed. If you use '/'' instead of '@', an new result page will start from this treatment row.

For rows with '+' or 'F', it is possible to add a number 1-9999. When printing labels or analysis forms, these numbers will be matched against the same number in the text template (but only if the template key contains any number) used for printing. These templates are stored in table FD46. See also *chapter 3 & 5*.

You can also add extra text rows to the treatment list. They should be numbered so that they come after all original rows with treatments. Extra rows will be printed after the preceding treatment rows in the order the come in the list. However, if a row number is entered in the TP LS column, this text row will be printed before that given treatment number. A '/' in the TP LS column will also insert a page brake.

TEXT: Here the explaining text for each treatment is given. These texts are important. They are used for printing results but also for data retrieval based on search criteria on used treatments.

Create field distributions of treatments, menu Trials-Randomize

The next step is to create treatment distribution patterns for the new trials in the trial series designed above. The program can randomize treatments for the following designs: CR (Completely Randomized), RB

(Randomized Block), SP (Split Plot), SSP (Split-Split Plot), LC (Lattice Alpha designs, one factor), RC (Row-Column, one factor), RL (Resolvable Latinized, one factor), SB (Split-Block, 2 factors), SPB (Split-Plot-Block, 3 factors) and SBP (Split-Block-Plot, 3 factors). Other designs must be randomized separately and manually entered or imported into the database.

Randomization of incomplete designs LC, RC and RL presumes that arrays are available in the R section of the general text library (table FD46, see also *chapter 3*). The transport file 'Example.txt' contains such arrays and can be imported into the database. At present, the incomplete arrays support only one-factorial designs.

If you wish to exactly choose a certain pattern from the available arrays, you must change the design code in Series table, FD21. If you did select the RL type, but wish to specify that the RL0703 array (for 21 treatments), then just enter RL0703 instead of just RL.

To randomize the new trials you must fill out the dialog form. You may SERIES, TRIAL-ID, LOCATIONS, YEAR, CROP and TREATMENTS. The selection is activated only if the corresponding box is checked. You can also specify the sort order, which is often important when you are printing lists. For randomization it is best to have the SERIES identification as the first sorting field. See further details in the beginning of *chapter 5 and 8*.

Randomization options: The form has an array of check boxes for up to five treatment factors, F1…F5. If no boxes are checked, randomization will be done without restrictions. These options can be checked:
I_I Don't randomize first replicate for this factor (treatment comes in the order of treatment list)
I_I Don't randomize this factor in any replicate (comes in order 1-2-3 in all replicates)
I_I Repeat the same randomized order for this factor to the other replicates
I_I Rotate randomized order for this factor one step in successive replicates

The last box, 'skip one level in last replicate', will randomize one level less of the F1 factor in the last replicate of multi-factorial designs.

Incomplete small groups: When randomizing types LC, RC and RL, a group (incomplete block) number will also be included in the 'GRP' column of the created treatment distributions (stored in table FD35). Other designs normally not require this information. You may, however, entered such numbers in table FD35. Row number are multiplied with 100. Row 5 in column 3 will have the GRP number 503. This information can be retrieved into statistics analysis files with the @F8 and @F9 functions.

Storing/changing field distributions manually, menu Trials-Distribut

In this table the treatment distribution is stored for each individual trial. It is not sufficient to store the distribution for each location since distributions may sometimes have to be altered throughout the years in a long-term trial. The PLOT/TRM/GRP columns show the plot and treatment number (sequentially numbered from 1 and up). Here you can view or change the result of the randomization in the previous step.

Old trials that have been randomized in another program must be entered manually into the database. Maybe other trial distributions or templates should be copied to a new trial?

New distributions are best entered by pressing the NEW button an selecting the actual trial identification. An input form appears. The 'codemask' box up to the left should indicate how the treatment should be entered (the character for the first level in each factor is the most common way). **Note:** This input is case sensitive - "a" is not the same as "A". Enter the treatment code for all the plots and STORE.

The INITIATE button can be used to empty the form. The ROW-INS button inserts a row before the point where you last clicked with the mouse. The ROW-CUT will instead cut that row.

You can also change an old distribution with the NEW button. First find the trial-id to change. Then press NEW and check the box 'Load form with …'. Change , STORE and CLOSE the input form.

Series designs that couldn't be automatically randomized by the program, must here be stored manually. This is also the case for older trials that you wish to store in the database (as they already have a fixed distribution). To enter the new distribution, first remove any existing old pattern with the CLEAR button, then press the NEW button. Choose the specific trial from the list (the program needs the print mask for the field codes from the series design table - if it is empty some data in the FD31 or FD21 must be missing).

Completing trial background information, menus Trials-Places and Trials-Trialinfo

When the new trials has been laid out, you can continuously add all kinds of background information about each trial.

Information about the location is completed in menu Trials-Places. Here you may add name/telephone and address to the host (farmer/experimental station) for each trial, it's geographic location and type of soil. If you have an image file with a map of the trial, you can refer to it here. The standard size for the map is 200 mm width and 70 mm height. If you add ,S or ,A to the file name, the printed/displayed picture will match the given area in the input form. Many image file types can be used. See also *chapter 9* and the @M function in *chapter 3.* **Note**: To be able to view the image in the input form you must press the PRINT button. If you don't cancel the print, the record will also come on your printer. To see the filename again - press button > and < to get out of, and back to, the present record.

Other background information is entered from menu Trials-Trialinfo. Here you can store any information that is not covered by the TRIALS, SERIES and LOCATIONS tables and is not plot observations nor treatment averages. These values are stored in other tables - see *chapter 6*. Example of general trial information: Date/amount specification of planting, fertilizing, weed control, level of nutrients in soil etc.. All types of information should have an information code (see below). The INFO LIST button allows you to view and pick those codes that you have in your variable list (the menu Series-Var.codes). The encoding rules was discussed in *chapter 3.*

During the season, different trial information is stored when they are available. Enter TRIAL-ID, information CODE and specify what you like. If the information code has a corresponding @B code -see *chapter 3 "Basic information"*- it can be fetched for printouts and/or for statistics files.

Storing treatment specifications, menu Series-Spec.trmt

This table, FD23, is important as it is used for several functions:

1. Storage of treatment factor and level texts. Key is SERIES identifier. 'TRM' is also less than 0.
2. Storage of texts for results of statistics analysis. Key is TRIAL-ID and 'FAC' is always 0.
3. Details on applied treatments. Key is either SERIES, TRIAL-ID or LOCATION, TRM/FAC are both>0

The third alternative is described here. If treatment details have been supplied, they can be used as search criteria when retrieving data, as variables or factors in statistical analysis etc..

SERIES/TRIAL. You can use the series, trial or location key depending on set where the entered specifications are the same. If you wish to specify a 'normal Nitrogen' treatment level in kg N/ha you can do this is this table. The TRIAL identifier will normally be used since the applied levels will be different from place to place and from one year to another ('normal' being set in practical farming by the field personnel).

TRM/FAC: You have to enter both the treatment number and the factor of this treatment for which the level is specified. TRM can be one of the existing F1…F5 factor (1-5) or any number between 1 and 99. The treatment description can be divided in as many subparts that are needed.

The columns VALUE/LEVEL, UNIT and TREATMENT text is filled out with appropriate details.

The @S function in *chapter 3* describes how the specifications can be fetched from the database.

It takes some work to enter these specifications. When you have done it once, you can use the COPY button to edit a copy for another (new) series or trial.

Data collection orders, menu Trials-Other

You can prepare a transport file listing, for example, trials, variables and sample numbers to be analyzed at a laboratory. You must specify the variables in a result profile (max. 10) and as type enter P, X, H which means samples for all plots, treatments and levels of last factor. The type S is for treatment samples requiring text data as result.

Before the transport file is created you will be given the chance to edit your order. A form will appear with pre-selected samples. Example: For the trial T-01 20 plot samples should be analyzed on moisture content (variable code MC). In the form appears for this trial under the MC column the text “P1-20”. If you now wish to analyze samples only for each treatment and, in fact, only for treatment 3,7 and 16-20 then you would change the order to “X3,7,16-20”.

After your changes, press to OK button to store the transport file. This file will be stored in the job log under the name 36<user>.job.

5. PRINTING OF LISTS AND FIELD CARDS

Preparation and selection of trials

Before you can print list and field cards, you must **define what to print**. Texts for labels, analysis forms and diaries are specified in the general text library, menu Printouts-Printlib. The way to do this is described in *chapter 3*. Texts for the field card must be designed in the menu Series-Fieldcard. The table under menu Series-Observat. is then also used to specify different kind of observations and registrations to be done in the trials. This is also described in *chapter 3*

Besides designing the layout/texts you must **set a printer and the fonts** to be used. This is done in the File-Setup menu. See also *chapter*. In order to get nice printouts, there are certain requirements on the used fonts (font number 1 to 5 as described in chapter 9). Font no. 3 and 4 should fixed fonts that prints at least 12 and 16.5 characters per inch respectively (point sizes - typically 10 and 7-8). Font no. 2 can be proportional and be of a size that will print about 10 characters per inch. An example is Times New Roman, size 14. Font no. 5 is only for printing bar codes. You should use a type of bar codes that allows the use of alphanumeric characters like Code 3of9. The density must be so high that approx. 8 characters per inch can be printed. Different lists and printouts use different default fonts as described below. The font no. 1-5 can be altered by the ^ function described in *chapter 3*.

The selection of trials is done in a similar way for all the printouts in the ”Printouts” menu group. You may specify year, sub-year, and any interval of series, trial-id, location or crops. Only the intervals with the corresponding marked box will be active. You can also set the order to print the selected trials. Often you use the order series - location - trials. Since the location key is often connected with regions in your country, one often use that key as first sorting parameter.

Field card, menu Printouts-Fieldcard

Field cards is a kind of check list or summarizing paper for the laying out and management of each individual trial. The different possible prints for the field card was described in *chapter 3*. They are:

a) File with full project description/guidelines (if specified under Series, Design)
b) Front page with treatments, map, observations
c) Basic information & variables to register (Bar codes)
d) Protocol for manually entered variables (blank columns)
e) Protocol for manually entered variables with printed variable codes on the protocol
f) Treatment text codes (meant for printing them as bar codes)

You mark the boxes for the printout you wish to get. All the pages produced from items b-f use font ^3 by default and font ^5 for bar codes on page c) and f). The text font can, however, be altered with the ^ function. If you are not using a fixed font, you must use the ^T function to insert a tab. This means extra work, so it’s recommended to use a 12 chr/inch font.

The printed text, as default, have series or location code as their key. With the buttons FIELD LAYOUT and OBSERVATIONS other text groups may be selected and used.

Labels, menu Printouts-Labels

Designing labels was described in *chapter 3*. In this menu sample labels (which can also contain bar codes) can be printed. Select trials as for field cards. Enter, or select with LABEL TEXTS button, a code for the label layout. Mark what you wish to get:

a) SAMPLE LABELS, labels for individual samples
b) BIN LABELS, for packaging samples, 2 labels/24 individual samples

Choose if you wish to print labels for 1) all plots, 2) all treatments or 3) First F1 level of treatments. The last alternative prints only labels for treatments with the first level of the first (F1) treatment factor. This is often the same as an untreated block treatment.

Choose to print 2*8 or 3*8 labels per A4 label sheet. If you wish to make page breaks (PAGEBREAK ON KEY) when your sorting key changes, enter the number of characters from the left in the sorting key to check for changes (producing the page break). If your 1st sorting index is locations, and your locations code always start with a 2 character region code, you might enter 2. You will then get a page break when the region changes. Then labels for different regions will come on separate label sheets.

If COPIES is not given, you will get a number of copies corresponding to the no. of harvests given in the main, Trials-General, table. The data load function @N1 can be used to print the copy/harvest number on the labels. You can also use the FREE CODE field to enter your own treatment codes. Separate the codes with a comma. To print them on the sample labels, you must use the @C function (see *chapter 3*).

Analysis forms, menu Printouts-Analysis

With this choice you can print prepared analysis forms. The default printing font is font ^3. These forms may have a header and a data section where data can also be entered manually. Select the layout with the LIST TEXTS button. The design of layouts was described in *chapter 3*. The COPIES and FREE CODE field are used as for labels. For analysis forms you can mark the box PLOTS WITH TREATMENT if you wish to print the plot numbers that has each treatment combination. This function only works for treatment listings and not for plot data forms. The information is useful if one should first put together a treatment sample from plot samples. You can also choose not to print treatment codes. That can be used for making standard (dummy) forms, not connected to any special trial (that information could be requested on the form).

Diaries, menu Printouts-Diaries

Diaries (check lists) is useful to keep track of the status for each trial. Which trials do you have? What steps have been done so far. What trial documents exist in my archive?

The design of diaries was described in *chapter 3*. You can get a cover sheet if you have used negative row numbers for designed texts. The diary prints a line for each selected trial. The content of the line is composed by the given texts and @ functions on rows with positive row numbers. The with of each data column in the diary is set by the = character. The text before ”=” is used as header, the text after ”=” as data (normally a @ function).

Before printing , enter the DIARY TEXTS code. If you mark the condensed print box the list will be printed with font ^4 instead of ^3 for normal print. You can also select to print the diary in portrait or landscape orientation.

Job lists, menu Printouts-Joblist

For the field personnel it can be handy to check what should be done in their trials at a certain time or growth stage. Such job lists can be printed. However, this require that you have specified information about a job in the table under menu Series-Observat. This table is also used to print information on the field card front page - the lower parts (part 3 and 4). Details can be found in *chapter 3*. The routine also looks for raw data files (only files with a name extension) in the raw data directory specified in the database settings (menu File-Setup). If it finds a file, a # sign will appear. **Note**: If a job has been given a ”job code” the raw data file should have a name beginning with the trial-id followed by the job code. Since the database is not adopted for Windows-95 file names, the length of the ”trial-id” + ”job code” should not exceed 8 characters (trial ”XX1234” and ”G1” can form a valid name ”XX1234G1.<ext>”.

When printing a job list, first make a standard selection. If you don’t select an alternative OBSERVATIONS list the program looks for a list with series/location/trial-id as key. Normally you also wish to search for an interval of time (mark box INTERVAL DATES) and/or of growth stages (mark box INTERVAL GROWTH STAGE). If no intervals are given all D- and G-marked rows (so marked in rows with POS=3 or 4 in the general text library, see *chapter 3*) will be printed in the job list. Preferably use decimal (0-100) growth stages for all crops. When selecting on growth stages it is often required to select also a crop type among the standard selections. Date intervals should be entered in the form MMDD in order to form valid intervals (both in observations table and in this search form).

6. HANDLING OF COLLECTED RAW DATA

Even if you have collected most data on data collecting field/laboratory computers, you often need to enter some data directly into the database. This is done in the tables FD74, FD75 and FD76, menu alternatives Base, Averages/Plotvalues/Text-Stat. Treatment values should be stored under the same treatment numbers as were used in the treatment list (menu Series-Treatmnt). Plot values should be stored in plot number order. Text-Stat data are stored as treatment averages – under the same numbers as in the treatment list.

Copying or importing data variables

The input form for both the menu choices look alike and you work with the data in the same way. Do you wish to **copy** data from one variable to another? Click on a row with the variable to be copied and then use the COPY button to store it with another variable code (the copy button works here on the variable within the trial - not on the leftmost column as usual). Copying data from one variable to another can be useful when you have assessment values that, for example, should be stored for many observation times and there are only minor differences between these times. You can then easily change a few values in the copy. If you use the copy button, the variable is automatically copied also in the list of registered variables (see below), but **note** that you might have to change the registration date in that table for the copied variables.

You can also **import** data from a spreadsheet file, database table, or ASCII file. The SPREADSHEET button can import data from spreadsheets with the structure given in *Appendix B*. There is, however, a problem to import Excel data with more than two decimals. Furthermore you can only import from the first sheet of a working book. To import data from other databases or text files, these files should have the same fields as the database tables. See *Appendix A*. The treatment average and plot value tables are named FD72 and FD73. Text files should have the field separator that you set in the File-Setup menu. They are imported from the File-Import menu.

Storing new data

Use the NEW button. **Do not enter new variables by simply changing trial-id and/or variables or by adding data in the end of the table.** Data entered in this way will not match the stored information in the variable list! The NEW button opens a special form for entering new data. Start by entering the trial-id (TRIAL), variable code (VAR) and registration date (DATE). Note that the date must always include the year.

The AREA field is used to store this information in the list of registered variables so that data can later on be re-scaled to figures per m2 or per hectare. This is often needed to be able to compare yield figures. If you have raw data for, for example, a plot with a net harvested area of 12 m^2, enter 12 in the area field.

The DEC field allows you to skip the decimal point when you enter decimal numbers. Enter the data without the decimal. In DEC you enter the number of digits that are to the right of the decimal point. If DEC contains '2' the data 4567 will be stored as 45.67. You can also enter a negative value in DEC. Data will then be divided by 10^{DEC}. If DEC is -2 the data "78" will become 78/0.01=7800). You can skip entering unnecessary zeroes.

Now enter your data. Remember that treatment averages refers to the treatment list number 1 and up, plot data to plot numbers from 1 and up. Data is stored with the STORE F9 button. When you take the next trial/variable, don't forget to always correct the information at the top (TRIAL ... DEC)

There are some other useful buttons below the data part. The INITIATE button is used to clear the data form (enter a space as initial value). You can also start with another value. Then you must add a comma and the number of rows to be filled. For example, to fill the value 100 in 30 plots, enter 100,30 as initial value. Use the INSERT and CUT buttons to move your data up and down (if values have come on the wrong rows). To CUT: First click on the row to be removed. To INSERT: First click on the row before which a new row should be inserted.

The NEW button can also be used to load existing data. First find the data and then, when pressing NEW, check the box option "Load …". You can now make change and replace/save this variable.

You can use the RECALC button to recalculate a store variable. Syntax – see operation C in *chapter 7*. You can use conditional recalculation also. For example, the formula "/10 IF>100" will divide data by 10 only if the data is greater than 100.

Variable list, menu Base-Varlist

This table keeps track of what variables have been stored for each individual trial. The first of the two TYPE fields contains the origin of data: + for original data, C for derived data (calculated with the database recalculation routines. The second TYPE field is X for a treatment average variable (stored in table FD72) and P for plot values (stored in table FD73). Registration date and observed net area (if applicable) is also stored. The SELCODE field is seldom used. If it has a negative value, that will effect the printing of factor averages for that variable. For example, if selcode is -1 then only factor averages for factor F1 will be presented on the result form. **Note**: If you print columns in date (D) format - see *chapter 7* - no factor averages are printed (averages of a set of dates in the form MMDD - like 0810 for August 10 - is often not meaningful).

Raw data handling programs, menu group Rawdata-

Under the menu group Rawdata, calls can be made to various routines for raw data collected with computers. The raw data collecting system can be built up by MS-DOS or Windows executable programs. *Appendix C* has more details. The following typical routines can be activated from the menu. The database users are referred to *separate manuals for the data collecting system.*

PC-server. This choice is intended to call a program that sets the PC standby waiting for incoming data or to transmit some data. Data collecting computers may be directly connected to a PC via a serial communications port or PC card. Data could be received via a Hayes compatible modem standing in auto-answer mode.

Receive. This may be used for receiving data via a serial communication port, by copying data from floppy discs or in another way.

Decode. When you have collected and linked the raw data for the trials, you could use this alternative to call a program that views and ask for changes of this data. The program can then output data on a form that can be imported to the database tables (the tables FD73, FD74 and FD75).

Store. If data is to be stored with this menu choice there must an input file named 56<user>.JOB. This file should have the format of a transport text file with the correct field separator, decimal character and date format. You can optionally have negative treatment numbers for records to be stored in table FD73. This will tell the database that these treatment average data rows comes in the order of the plots in the trial and not in the order of the sorted treatment list. If you import data as a transport file from the File-Import, negative row numbers will be stored unchanged (you must the correct them manually - to corresponding, positive treatment numbers).

Using treatment codes in data collectors, menu Trials-Fetchdist and Trials-Senddist

The menu choice **Trial-Fetchdist** can be used to create a special ASCII file with general trial information and the treatment distribution for each selected trial. The found data will be put in the job log file, 36<user>.JOB. The amount of information you get in this file is very limited. If you wish to produce a more complete list of data for each trial, you can take out a so called ”Transport file” from the Base-Retrieve menu. The layout of the treatment distribution files and the transport file are described in *Appendix B*.

The menu **Trials-Senddist** can be used to call a user-defined routine (MS-DOS or Windows program) that downloads treatment distribution files or more complete trial information to the field/laboratory data collecting systems.

Linking of field data, menu Rawdata,Link

This program links together data from transport files. These files may come from other databases, from laboratories or from field data collecting devices. They may have been created through the first two items in the Rawdata menu. Transport file format is described in the separate file *Dbtransp.pdf*. Data are merged into so called FDL files (the files filename extension), one for each trial identification. The files are text files with data in columns. Only data for tables FD33 and FD73-FD76 (that is, raw data) is converted into the files. The files can be printed, or they can be viewed and re-coded with a Decode program (called from Rawdata, decode).

To link the data in the best way, the database settings (File,Settings) should be set so that,

- New incoming data are stored in the folder Raw data files, new
- All incoming data for the season are stored in folder Raw data files, all
- All linked data, FDL files, are stored in folder Linked raw data

When new data is available, it is first linked into FDL files in the 'New raw data' folder. The incoming data is then moved to the 'All raw data' folder. Then this folder is converted into FDL files stored in the 'Linked raw data' folder. Decoding of data normally takes FDL files that are temporarily stored in 'New raw data' folder.

There are some special settings for the linking/tabulating of field data. See ADVANCED SETUP in *chapter 9*. One option (default) is to have treatment averages calculated at the end of the FDL files. This function requires that the treatment distribution is found either in the FDL file or is stored in the database. If net areas has been given for some data variables (like net harvested plot in sqm), the treatment averages is re-scaled to the m2 unit, however if the net area was larger than 10 m2, then the averages are presented per 100 m2.

Decoding of linked data, menu Rawdata, Decode

This extensive program option can be used to decode linked FDL files before they are stored in the database. The program is not described here since, outside Sweden, raw data files will normally be transport files or be in another national data format. An English version of the decoding program will be available later.

The present, Swedish, decoding program is a MS-DOS program. It reads through each FDL file and lets you manipulate the data, change variable codes and so on. There are nine basic program parts:

1. **Remove variables**
2. **Add variables**
3. **Adjust number of rows (plots)**
4. **Merge or copy data columns**
5. **Make sums or averages, vertical or horisontal**
6. **Replace data items**
7. **Roll data (move between plot/row numbers)**
8. **Printouts (temporary)**
9. **Save to transport file + next trial**

7. ANALYSIS AND PRINTING OF RESULTS

When you begin to get results from the trials, you would like to run statistical analysis and print the results. The statistical analysis is often done both for the individual trials and for series of trials. This must be done with a separate program but the database has routines for extracting data for your statistical package. You can also prepare some routines/programs that process the output from your statistical package and prepare for storing some statistical parameters in the database. The programs data retrieval part is often used to get data from series of trials, for extracting data for making diagrams in spreadsheet programs and so on.

The following steps must normally be done during the result analysis process:

- Composing calculation formula sets
- Composing result presentation profiles and statistical extraction profiles
- Making calculations and extracting statistics files
- Running statistical analysis and completing trial data with statistics results
- Printing of results, making data searches for presenting diagrams etc..

This seems to be a lot of work - and it is. However, the first two steps can often be done earlier in the season in order to speed up the process of getting the results.

Composing calculation formulas, menu Outputs-Formulas

Here you have your library with calculation formulas. A set/group of formulas have the same FORMULA key. By default you can use the series code as your FORMULA name for each trial series but you can use any other name (and select that name when the calculations should be run. In the ROW column you number your calculations in the order they should be run. You should use row number 0 as a title row. Put the title in the TEXT-COMMENT column.

In general, only treatment averages (table FD74) and plot values (table FD75) can be trecalculated. Text-Statistics data (table FD76) can only be retrieved for result prints and statistics files.

The OP means operation and defines what should be done. There are about 10 defined operations described below. Other OP will be treated as a comment. You can write your own comments in the TEXT-COMMENT column. You can see below what will be calculated when you enter different parameters in the VAR1, VAR2, PAR1 and PAR2 columns. OUTVAR is the variable code for the result of the operation (for those operations that creates new variables). If the result variable already exists, it will be replaced. However, original data will not be replaced with confirmation by the user.

Most operations can, according to certain rules, also be performed by using wildcards in some of the fields OUTVAR, VAR1 and VAR2. This can be very handy sometimes, for example when calculating dry matter yields from several harvesting times in forage crops over the season. The PAR1 and PAR2 fields can contain @ function that get numeric parameters from the database (@A5 is the harvest year, for example).

The sign * means ”any text” from the position where the star is and should therefor only be used last in a given code. The ‘?’ character means ”any character in this position” which is more useful in many operations. The patterns [xy] and [^x^y] respectively means that a variable code should or should not have the character x or y in this position. Below given syntax has a [?] where wildcards can be used.

OP=	**+ - * /**	**Addition, subtraction, multiplication, division**
+	Addition	OUTVAR[?] = VAR1[?] + VAR2[?] + PAR1
-	Subtraction	OUTVAR[?] = VAR1[?] - VAR2[?] - PAR1
*	Multiplication	OUTVAR[?] = VAR1[?] * VAR2[?] * PAR1
/	Division	OUTVAR[?] = VAR1[?] / VAR2[?] / PAR1

VAR1 should always contain a variable code. VAR2 can contain another variable code and/or PAR1 a numeric constant. At least one of VAR2 and PAR1 must be given. You can use @ functions under PAR1/PAR2 to fetch values to include in the recalculations. A special case is the @V6 (registration area for a variable) and operation “+”. If VAR1/VAR2 are to specific variables not only the variables will be added but also the area from @V6.

The database keeps track of the type of data to be calculated (in table FD73). The resulting variable will normally be of the same type as the in-going variables (VAR1 and/or VAR2). If both VAR1 and VAR2 is used but has different types (that is one is plot values and the other treatment averages), the program will combine each plot value with the right treatment average (by using the treatment distribution table). The resulting OUTVAR will be of the plot data type.

OP= = Assignment

The operation assigns data to OUTVAR from VAR1. The assignment is written in PAR1 and/or PAR2. Example: OUTVAR='DM' (dry matter %), VAR1='MC' (moisture content %), PAR1='100-@' will be interpreted as DM=100-MC. The symbol @ stands for the VAR1 data.

OP= A Addition

The operation works as "+" above when wildcards are not used. However, OUTVAR is created even if one in-going variable is missing. If wildcards are used the syntax is: OUTVAR = VAR1[?] + PAR1

OP= B Blocking

The function sums sub-plot observations that has been input vertically, that is, for example, subplot 1-3 is from plot 1, subplot 4-6 from plot 2 etc.. PAR1 may contain a re-scaling factor. PAR2 should contain the number of subplots (3 in this example). If PAR2 is negative, the B function calculates the average of the subplots instead of the sum.

OP= C Calculus (free formula)

You can use this operation to recalculate data in existing variables with a user-specified formula. A variable (or as set of variables using wildcards) is given as OUTVAR. The fields VAR1, VAR2, PAR1 and PAR2 are all used for the formula. The OUTVAR variable is denoted @ in the formula. Any conditions should come after "IF". Example: The combined text VAR1…PAR2 reads: "(100-@) IF >0 AND @<=50". This will result in a transformation of (for example) water contents to dry matter contents in grains if data appears to be in water percentages. **Important:** When changing data this way, you often use the N operation to rename the changed data so that the same operation cannot be run a 2nd time (OUTVAR no longer exists).

OP= D Delete

Deletes data for variables in the interval from VAR1 to VAR2. If you use wildcards, give this in VAR1 and leave VAR2 blank. OUTVAR, PAR1 and PAR2 are not used.

OP= E Exponentiation

The function raises a base given in PAR2 with the data in the exponent. If PAR2 is not given, the natural logarithm, ***e*** (approx. 2.71828) is used. If PAR2>0 then this number will be used as base. From the calculated result is then *subtracted* any constant in PAR1. PAR1 is often set to 1. See also the reverse function L below. The result is stored in OUTVAR. VAR1 and VAR2 are input data. If VAR2 is given the sum VAR1+VAR2 is used in the calculations.

OP= G General trial information

This operation writes a control list for each trial to the job log. This list contains numbers of treatments, plots, stored general trial information, stored average and plot variables etc. Together with the use of the M function you will probably have enough information to judge whether or not the data for each trial is more or less complete. When you use only the "G" operation in a formula you can enter -1 in the formula header row (row number = 0) in order to get a more compact job log with only "G" data (one row per trial).

OP= J Julian dates

Recalculate date figures in variable VAR1[?] to date numbers. The variable data can be stored with or without the year. Dates within on year can be stored as integer numbers for the combination month-date. August 1 should be entered as 801, October 15 as 1015 etc.. If you include the year, it should be entered as decimals. Example: July 25, 1996 is entered as 725.1996. You add a reference date with month/date numbers (like 0801 for August, 1) in PAR1 and a reference year (like 1992) in PAR2. You can also enter a @ function in either PAR1 or PAR2 that will fetch a reference date, for example the planting date. The J function will recalculate the following data with the reference 0801 and 1992 as follows. Year-less date figures before the reference date are supposed to refer to the next year - which is normally the year of the harvest! 801 => 0 815 => 14 725.1991 => -372 731 => 364 (July 31, next year - 1993)

OP= L Logarithm calculation

The function calculates the logarithm of input data. This technique is often used to create a more normalized data material for statistical analysis. The most common trick is to calculate the natural logarithm, *ln,* from input values added with 1. Enter one or two input variables under VAR1 (and VAR2). PAR1 can hold a constant, normally 1, which is added to data before calculating the logarithm. If input data is 0 then the result will also be zero, if 1 is not added, the result will be infinite negative in this case. If PAR2 is not given, the natural logarithm is calculated, otherwise the PAR2-logarithm is calculated.

OP= M Matrix print

Prints all replicates of all treatments to the job log file 61<user>.JOB. Enter an interval of variables from VAR1 to VAR2. In PAR1 you can specify a certain treatment factor 1-5 (0 gives all combinations). PAR2=1 prints variables with averages as well. OUTVAR is not used and wildcards cannot be used.

OP= N Name change
This operation just renames the variable(s) in VAR1[?] to the OUTVAR[?] name(s)

OP= P Plot values
This operation just moves data from treatment average table to plot data table. This is useful if plot data was stored as treatment averages. If OUTVAR<>VAR1, the variable code is changed. If PAR1 and/or PAR2 is given, data will be re-scaled.

OP= R Relative numbers
Creates relative numbers in OUTVAR[?] for variable(s) VAR1[?]. Enter treatment factor in PAR1 (1-5). If PAR1 is 0 or blank, relative numbers for all factors will be generated. PAR2 can contain the level of a reference treatment - if not entered, the first treatment is used as reference. When generating R for all factors, the PAR2 field can contain f=nnnnn where you replace f with a certain factor (f=0 means the lowest factor) and nnnnn with the reference level for that factor. In variety trials, where the variety number is a part of the treatment code, that variety number should be used as reference level.

OP= S Subtractions (relative numbers)
This function works as R but instead of % figures you get the absolute differences from the reference.
Note: The difference from two rounded figures on a result print will sometimes differ slightly from the more exact differences produced by the S function on the not rounded stored data.

OP= T Treatment averages
If you wish, you can store treatment averages (under another variable code) for plot data. Enter the plot data variable(s) as VAR1[?] and the treatment averages as OUTVAR[?]. VAR2, PAR1 and PAR2 is not used. Averages from plot data with missing values is adjusted according to the method specified in the series design. **Note**: You don't have to generate averages for printing results. It is done automatically (not stored).

OP= V Variance table
For simple 1- and 2-factorial trials, this function can create a simple ANOVA table and calculate Fischer LSD-values. The Anova table is printed to the job log only. Result data is stored in table FD76 under the same variable code but with the character ‘ added at the end (to distinguish it from the original variable). In the PAR1 column you should add a factor if the Y variable (given under VAR1) is re-scaled before printing on result form. The factor PAR1 will then re-scale the Anova-table data and LSD-values accordingly.

OP= X Move plot data to treatment average table
This function moves data from the plot data table to the treatment average table. Unlike the T function, data order is not changed. See also the reverse function P above.

Result profiles, menu Outputs-Profiles

This is a library where you can specify what data to take out to statistics files or for printing trial results. The profile key can be named after the series code (default) or have any other name. Under numbers under COL specify the order of the output data columns. Use col=0 for a short profile title (put it in text row 1-2).

The button VAR.LIST can be used to pick variable codes from the library (table FD25). You can use the EDIT MASTERS button to view and change different master files for the printing of results. Master files (templates) can be edited also in a MS-DOS text editor. See *chapter 3* for more details.

As variable codes you may also enter wildcards. If you wish to get the total yield and yield for different harvesting times in a forage trial you may use the code Y? where Y0 could be the total yield and Y1..Y9 the yields from different harvesting times (max. 9 in this case). In the text explanation fields you may put question marks where you wish matching variable code parts should be printed. If you have the text ”yield ?” it will print yield 1 for Y1, yield 4 for Y4 etc..

Printing results. For printing of results only variables can be output. Codes is entered under VARCODE. If the TYPE field is blank that variable will be included only if registered. In the TYPE field you can specify:

+	Always include this data column in the result (regardless if the variable was registered)
/	Make a page brake before the following data columns. Should not contain any variable.
-	Never include in result forms. Only for use when extracting statistics files.

For extracting statistics files, the TYPE field can also contain:

Y	Dependent variable to be analyzed
F	Fixed independent factor effecting the Y variables
I	Information variable (not directly processed in statistics program)
R	Random independent factor (like replicate/block numbers)
X	Continuos independent factors/variables effecting the analyzed Y variables
S	Sorting/grouping variable which may be used when analyzing statistics

The DEC field is not used for statistics files. For result forms you enter the number of decimals for the data in the result column. If you enter a negative number values will be rounded: -1 rounds last digit, -2 two last digits etc.. If DEC is D, data will be printed as dates (801 will be printed as 01-08).

In the COLUMN TEXT ROWS (not used for statistics files) you enter the texts you wish to appear above each data column when printing results. The column has place for 8 characters. How many characters you can use depends on the column width you choose when printing the results. If you, for example, have chosen 5 characters wide columns, the column texts cannot be longer. You could enter one extra character that will appear between that and the next column. If the extra character is a ^, it will be printed as blank.

You can use ? where you wish matching part of variable codes to be printed (if you are using wildcards for variables as described above). If you wish to print the registration date, plot area etc.. you can enter @V5, @V6 in the text row you wish (@V functions - see *chapter 3*).

It is sometimes convenient to multiply the stored data with a factor before printing it. An example is stored dry matter figures. For grains and seeds you might wish to present them at, let’s say, 10% moisture content figures. Then just enter the factor 100/90 under SCALE FACT.

For **statistics files** other data than registered variables can be output. You can load other data by entering a @ function in the VARCODE field. See details in *chapter 3.*

Calculations, menu Outputs-Calculate

When you have composed your formulas and result profiles, and have stored the data for some trials, it is time to analyze them. First you should run the calculations you need in order to get all your trial results stored. For example, you might wish to calculate clean dry matter from harvested amounts. Select trials. If your formula has another name than your series code, select it with FORMULAS button (or enter the name).

See above about formulas. All calculations is made directly on the data in the database. You can create new variables or replace existing data. If original data is to be replaced, you must confirm it. A job log from the calculations is stored in the log file 61<user>.JOB.

Statistical analysis, menu Outputs-Stat.file

Choose this menu to output data to ”input” files for your statistical package. You can put the data in one file per trial (1 FILE/TRIAL) or in a common file for all trials (1 FILE ONLY). Enter the filename of the statistics file(s). If you have chosen one file per trial, just enter a file name extension (like TXT). Each created file will be named after the trial-id and with the entered extension. **Note**: The database cannot handle longer filenames than 8 characters, so the trial-id you use should not be longer! For one common file you should enter the full file name.

If you specify the SAS filename extension, the database will generate a ready-to-run program for the Statistical Analysis System (SAS) package. The SAS program generator is further described in the document *DbandSAS.pdf* in the database folder.

The type of statistics file depends on the filename extension. Most extension results in a text files. However, extensions like XLS (Excel), WKS (Lotus), DBF (DBase), DB (Paradox) etc. produce spreadsheet or database files. It is also possible to take out statistics files from the data retrieval menu. See *chapter 7* for details about possible file types and *chapter 8* about data retrieval.

If you mark the box USE PROFILE you will get only the data specified in a profile (see above). Use the PROFILES button to pick any profile you wish. If you don’t enter one, your series code is assumed to be the name of the statistics profile. If you don’t use any profile, you will get files with all factor levels and all stored variables - but no other information..

For text files you can choose between some more alternatives for the data labels (variables). You can also specify if you want to have blank-separated (fixed format) files or files with a special field separator character between the columns. In the latter case you can choose the ASCII code for another separator than the one you have set in your database (suggested in the form).

Storing statistics results, menus Base-Stat.pars and Outputs-AnovaRes

When you have done the statistical analysis of your trials, you can store some result parameters in the database table FD75. You can do this **manually in the menu Base-Stat.pars**. Under TRIAL-ID and VARCODE you enter the trial identification and the code for the analyzed, dependent variable (typically the yield). As ROW number you are recommended to use 200 up to 250. The numbers should correspond to explanation texts listed at the end of the treatment list for each trial series (the rows where TP contains @S). As value you can enter any text, for example ** or 0.003 for a significance level. You should not enter more characters than the column width you use for your result forms.

You can also use the **Output-AnovaRes** menu, which is intended to call a user-written MS-DOS or Windows program that can **automatically** process the output from your statistical program and put in a file to be imported in the database. Such a file should have the 4 fields listed for table FD75 in *Appendix A* and the field separator you have set in the database File-Setup menu. See also *Appendix C* about program calls. Import such result files from the File-Import menu. For the statistics programs SAS and NCSS the program FD66.SBP has been included which can automatically take back data from the output files from this programs (SAS output from the generated program and NCSS Rich Text File Log).

Printing results, menu Outputs-Results

Here you output results from the trials. When you stored information about the series design, you also entered which printing master you wish to use for the data (the master STANDARD.RBL is default and comes with the installation). You can choose to output:

a)	RESULTS ->	PRINTER	Results is written to the printer or the screen
b)	RESULTS ->	TEXT FILE	Results is written to a text file
c)	TEXT FILE ->	PRINTER	A result text file is printed on the printer or screen

Normally, you print the data directly (a above). If you wish to manually adjust some results, you can output the data to a text file (b above), edit this file in a text editor, and then finally print it (c above). Stored text files are named after the trial-id with filename extension TXT. Since the database cannot use longer filenames than 8 characters, your trial identifications should not be longer than that.

If you wish to use a data profile, mark the box beside the PROFILE button (use the button to pick a profile). About profiles - see above. If you don't specify a profile name, the series code is assumed to be the name to use. If the box is not marked, all stored data variables will be printed.

If you wish to use another printing master (template) than the one entered in the Series-Design menu - select one with MASTER FILE button. How to edit the masters was described in *chapter 3*.

8. DATA RETRIEVAL

Searching data, menu Base-Retrieve

In this module you can select data from the database more freely. You can make up to 9 searches (step between them with < and > buttons). All entered criteria will be combined. Selections can be made on (besides general information) on background trial information, treatments (texts, specifications) and variable values.

When searching for texts, the asterisk (*) is used for any text. The search text *301* is used to look for ”301” anywhere in a text, 301* for texts beginning with ”301” and *301 for texts ending with ”301”. If you don’t use * the searched text must be identical. The text search is not case sensitive. Search for ”*abc*” gives the same result as ”*ABC*”. You can also use ”?” which means ”any character in this position”. For example, ”????-301” finds ”XDEP-301” and ”YDEP-301” but not ”XDP-301”. You can also select among alternative characters between the [and] markers in the following way. Suppose you wish to select trials who were located in regions A, F, G, H, R, S and U. This can be done by searching for locations ”[A F-H R-S U]*” which means any location codes starting with those letters followed by any text (don’t forget the ”*” sign for this). You can also exclude characters with the ”^” sign. If you wish to find all trial series except those who comes from XDEP and YDEP you can search for ”[^X^Y]DEP*”.

In the search ,as well as in the reduced for printing results etc., you can enter alternatives with comma signs in between or enter an interval of the form “from_text>to_text”. The (>) and (,) characters will always be interpreted as interval and alternative markers (a ? must be put in search patterns to avoid this interpretation of these characters). Finally, a leading ^ sign (not inside []) will perform “NOT” for this search pattern/interval.

Selection on basic trial information is done by selecting a specific code with the “INFO-CODE” button and by entering appropriate search conditions for that specific property.

Selections on treatments is done if you enter anything in the “TREATMENTS: TEXT” field. If you mark the SPECIFIED TREATMENT box, the search is done in the table with specifications instead of the treatment list. Then you must also give some from-to levels (for search only on text enter very low and high values that falls outside any data in the table)

When selecting on data values, don't forget to enter value criteria for the searched data. You must mark the INCLUDE PLOT VALUE box if you want to search also among the plot values (you need to do it if they are not also stored also as treatment averages in the database). If you use ?* as variable you will search for any variables that have data in the given interval, Y* all variables beginning with Y (like yield figures) and so on.

When you have many different selections they are all combined. If you need to find trials with either one condition or the other (and the , > and ^ characters was not enough in the upper part of the form) you must do like this: 1) make a search on the first alternative. 2) make the search on the second alternative and mark the OR box (you can first try the other alternative without “OR-ing” it with the previous search). The OR and AND options always combine the result of the present search (n) with the previous one (search n-1).

Outputting selected data

When you wish to fetch data from a selection step, with the < > buttons, to the search number you are interested in. Then press the OUTPUT button. Now you come to a form where you can select the kind of data you want and how it should be output: in a **window** (screen), to the **clipboard**, to the **printer** or to a **file**. If you want the data in a spreadsheet or database file, use the "file" alternative and enter a filename with an extension like XLS (Excel), WKS (Lotus), DBF (DBase) etc.. See below.

Basic info means the content of main TRIALS table as well as the SERIES and LOCATIONS tables. It is these tables that always must contain data for a trial. If a trial is missing in either of these tables, it cannot be found by the database search program. **General information** fetches all other information (from the Trials-info table) that is neither treatment registrations, nor plot values. **All data in selected trials** means all treatment averages and plot values.

Selected treatments will output only the matching treatments (if you have made a search on treatments). **Selected variable data** takes out the matching data rows that matches a search condition on stored data values (like ”Yield greater than x kg”). The **Selected treatments * var. data** means the ”common” section of data rows matching a combined search for treatments and variable values.

Transport files is a special file for transporting data from one database to another. It takes out data from all tables for the selected trials where there is a match on the TRIAL, SERIES or LOCATION code. This file can be imported into another copy of the database. For very large data selections, it may be easiest to export to a text file (spreadsheet and some other databases may have limitations on maximum field sizes or number of rows (the number of rows may then easily exceed 1 million). Transport files can be exported in scientific number and date formats. Dates will then be exported as Julian day numbers and decimal figures as a 10-digit integer with exponent. These formats more or less removes rounding errors and avoids problems with different date formats (like y2k problems). A special document on transport files is included in the database directory - filename Dbtransp.doc.

With the **series report** option you can take out data as a statistics file, but only of the types field-separated ASCII, spreadsheet or database files. You can take out either treatment averages or plot data files. It is also possible to extract data only for the levels of the last treatment factor (F-1). You must prepare and select a PROFILE. See *chapter 3* and *chapter 7* for details about profiles (result profiles) and supported data file types. The box 'SELECTED TREATMENTS' will only print matching a search on treatment texts.

Data file types supported by the date base program

The following data file types are supported by the database:

Filename extension	**Type of file**
$BF	Enable versions 1 or 2 files (if needed, rename it afterwards with .DBF extension)
DB	Paradox version 3.5 and earlier
DBF	DBase III and IV files (you can used DB2 as extension to get a DBase II file)
DIF	Data interchange format files, like VisiCalc etc..
LGX	Logistix files
SBF	Superbase file (separate from your own table)
SQL	SQL Server files
WKS, WK3	Lotus 1-2-3 files (the created file is converted to newer versions when loaded)
XLS	Excel version 3 or earlier (you can upgrade the version with your Excel program)

Note: If you export/import data to other Superbase files - **do not export to, or import from, table names used in the database itself (those in *Appendix B*, FDSYSTEM.SBF and FDHELP.SBF).**

Free search, menu Base, Queries

This search alternative can be used to make special selections from any database table(s). However, this requires some knowledge about "Superbase" way of data retrieval and also the information about table fields found in *Appendix B*.

1) Up to the left you can switch between the database tables to find the data you are interested in
2) Data from different tables these must be related through the table keys (ex. SERIES.FD21=ID.FD31).
3) These criteria are specified under "Filter" where you can also add other search conditions.
4) Under "Order" you can specify the order (field) to retrieve data. If the "Distinct" box is checked only unique data in the order field are output.
5) Select where to send the result of the query. If you choose "Answer" the result will go to the job log.

Free queries can be made complicated and you may often specify query conditions that will be wrong or take too much time to execute. The executing of a query goes much faster if conditions are applied only on indexed fields. The selections in the menu Base, Retrieve has been optimized in this way in order to ensure that quite large amounts of data can be retrieved within a reasonable time.

9. OTHER FUNCTIONS

Database settings, menu File-Setup

Here you set up the database so it will run properly. There are parameters to set how called program and printers should work together with the data base. Entered directories will be created if the don't exist. The directories are passed over to called programs according to *Appendix D.*

User & Password: Many users can be logged on to the database. However, they must all have different computer names. The database gives you a user name consisting of the first 10 characters of the network name for your computer. If it has no name, you will just be called Superbase. Those who have SYSADM rights (see about that button below) must also grant each new user name the right to open the database. In a network system it is advisable to use passwords. If you enter something in the password field - please remember it! You will be prompted for it the next time you start the database.

Organization name and Logotype file: Enter the name of your organization as you wish it to appear on lists etc.. If you have a logotype as an image file, enter the filename. You can add ,S or ,A to have the file adjusted to fit a given size. See also *chapter 3* regarding the @A and @L functions.

Path – general user work directory: The directory where you collect most of your temporary work files. You are recommended not to leave this path blank since your temporary files will otherwise be stored in the same directory as the database tables and system files.

Path - raw data files & new raw data: You can enter paths to directories that should hold the seasons all stored raw/field data and only new raw data respectively.

Path - linked raw data: Enter a path if you need a directory for linked (merged/sorted) raw data.

Path - result and statistics files: Here you should enter a path to a directory where you wish to have files for your statistics program. This path is also used for produced files from the database's search module (Base, Retrieve). See also *chapter 8*.

Serial comm. port, baudrate: These settings are used only for raw data handling programs for communications. The combined string of the two fields is passed on to called programs.

Field separator: Here you should enter the ASCII value of the character that, by default, should be used to separate table fields during export and import of data from/to the database. A special case is to set it 0. Then data will be exported/imported in a fixed format according to each table definitions.

Export/Import in scientific data format: If the box is checked the database will export dates as Julian numbers and real values as integers with an exponent part. With this setting problems with different date formats and different used decimal signs can be avoided and rounding of real values (when exported to a text file) is minimized.

The PRINTER button is used to select the **printer** to be used by the database. After selecting one, you will see its name and connection port (network printers should be connected to a port connection in your Windows settings).

The PRINTER FONTS button lets you define fonts etc. that the selected printer should use. You can store several font files. These file should always have extension .PAR. See *Appendix A* for details.

Langauge table type: Choose MS-DOS or Windows.

SYSADM functions: The one who installed the database the first time gets "System administrator" rights to use this button. This user can grant these right also to other users.

1=Grant new user is used to give a new user access to the database.

2=Revoke user removes all access rights for a user.

3=Change decimal character is used to define the decimal character to something else than a decimal point (like a comma). Note that Superbase uses its own decimal character which is not affected by your Windows settings.

4=Change language option the database language can be changed. This should only be done when no other users runs the database.

5=Edit help text is used by the system administrator to improve the on-line help text that the users can recall by pressing Ctrl-F1 in the program.

Printer settings and fonts

For the database to work properly, you must always select a **printer**. Use the PRINTER button in the File-Setup form. If your computer is not connected to a printer, you should install one anyway. The installed printer must also be connected to a printer port - even if you use a network printer.

For the different printouts, the database uses **font files with filename extension .PAR**. These files should be kept in the database directory. You change or create new font files from the File-Setup menu by pressing the PRINTER FONTS button. You come to a form where fonts and some other parameters can be specified. You can use the OPEN FILE button for opening another font file. To save a font file to another name - first change the font list filename at the top and then press SAVE FILE.

The LEFT MARGIN and ROWS PER INCH settings apply to printing of text files. The left margin is specified in millimeters and you can choose 6 or 8 lines per inch. The ^1 STANDARD font is used for printing text files. If it is a rather small font, 8 lines/inch is appropriate, else 6 lines.

The five yellow buttons with ^1 .. ^5 is used to select different fonts:

^1 STANDARD	Standard font for printing text files. Not used in printed forms, unless it is specified with ^1. It is recommended that the standard font gives a fixed number of characters per inch (fix font), so that it will print also space-tabulated data in nice columns.
^2 HEADLINES	Font for printing headers with larger letters. Here it is all right to use a proportional font, but it should not be print less than about 10 characters per inch
^3 12 CHR/IN	Use a fix font that prints12 characters per inch. This is the default font for printing field cards, labels, analysis forms, job lists, diaries and result forms. If a fixed font is not used here, texts in these lists can sometimes be structured with ^T (tab) function. However this takes time and doesn't work an all kinds of lists.
^4 17 CHR/IN	Use a fix font that prints 16.5 - 17 characters per inch. It is used for condensed print of diaries and result forms (but can be used in other lists by calling ^4)
^5 BARCODES	If you don't wish to print bar codes, leave this font empty. Otherwise you must use a font that has a density of 8-10 character per inch. The program is designed to print alpha-numeric data as bar codes. Examples of bar code types that allows this are: Code 3of9, Code 93 and Code 128. Mark the box "*text*" to automatically print asterisks around the bar code (Code 3of9 requires this as start/stop markers). You can also enter an ASCII code for printing space as bar codes (if other than 32).

In Windows, you cannot always see if a font is fix or proportional. For example, Times New Roman is proportional (size 14 is good for font no. 2 above) and Courier New is fix. The size 10 corresponds to 12 chr./inch for fix fonts, size 7 or 8 to 17 chr./inch. The mentioned fonts comes with the Windows installation and are True Type fonts. They should look the same on the screen as on the printed paper. It can often speed up the printing of forms if in-built printer forms are used, for example Line Printer (17 chr./inch) on HP printers. OCR fonts may be an alternative to barcode printing in some cases.

When printing all other things than text files, Superbase creates temporary printing forms. Besides selecting the fonts above, you can also set a relative page length and offset. The normal page length is A4 with some 7-10 mm top, right and low paper margins. By entering a value of less or more than 100 in the PAGE LENGTH field the form length will be adjusted to the given percentage of the normal A4 length. Entering a negative or positive number in the OFFSET field will move the printed forms to the left or right (normally with the number of millimeters entered). This is needed, because the database in a few cases doesn't fully comply with the installed Windows printer drivers.

Remember to save your font file before leaving the font setup form with the CLOSE button.

Picture/multimedia files: The program can handle the following file formats. However there are always differences in file formats so it is possible that some types cannot be read even if they ought to:

.AVI	Video for Windows file (sound)
.BMP	Windows bitmap-file
.CGM	Computer Graphics Metafile
.DRW	Micrografx Drawing file
.DXF	AutoCAD file
.EPS	Encapsulated PostScript
.FIF	Fractal Image Format
.GIF	Graphics Interchange Format
.IMG	GEM image file
.JPG	JPEG image
.MDI	MIDI - Musical Instrument Digital Interface
.PCD	Kodak Photo CD Image
.PCT	Macintosh Picture file
.PCX	PC Paintbrush
.TGA	Truevision TARGA file
.TIF	Tagged Image file
.WAV	Waveform file (sound)
.WMF	Windows Metafile
.WPG	WordPerfect graphics file

The button 'ADVANCED SETUP' is used to set some special settings.

PROGRAM DATA DIMENSIONING:

Maxium number of rows:	Maximum n:o of plots in any trial
Maxium number of columns:	Maximum n:o of variables in trials or reports.
Screen zoom, height & widht, %	You may enter a percentage to increase/decrease the number of rows/columns in some data input forms.

DEFAULT VALUES FOR RESULT FORMS:

Result column data alignment	Select right, left or centered data columns
Treament list, % width adjustment (100)	You can adjust between 50 and 150% (to better adopt the character spacing of used text fonts).

DEFAULT STATISTICS REPORT FILE:

Type of file with statistics results	For the moment only SAS (lst-files) or NCSS (rtf-files)

SAS STATISTICS PROGRAM OPTIONS:

Max factor interaction levels	Max. n:o of interaction levels in multi-factorial designs
Relative numbers shown when obs. >=	Display relative numbers if >= 'obs'
Treatment list blank when rows >	Add blank rows after each number of 'rows'
Formating string for reference	Format string for changing style of reference levels
Mixed iteration option(s)	Special SAS procedure Mixed iteration options
Mixed model option(s)	Special SAS Mixed model options

OPTIONS - TABULATED FIELD DATA:

No. of columns in data files	Max. n:o of columns in linked FDL files
PageBreak ASCII codes (code,code,,)	ASCII-codes for pagebreak in FDL-files
Include treatment averages in files	Include treatment averages in FDL files
Tabulate/link only new raw data	Only build FDL files for the new raw data

Exporting and importing data, menus File-Export and File-Import

You can export data from any of the tables in the list. You must then enter minimum and maximum value of the tables first/normal index. Enter the file name to export to. These can be text files or DBase-, Lotus-, Excel files etc.. See also the end of *chapter 7*. When importing/exporting text files the field separator in the database setup is used. As record separator (row terminator) is always used carriage return + line feed (ASCII codes 13 + 10).

When you import data in text files to selected tables, be sure that they have the right field separator (you can, of course, temporarily change it in the File-Setup menu). Database and spreadsheet files must contain the same number of columns as there are fields in each table. The data in the columns must comply to the data types and maximum string length in *Appendix A.* You can always export some data to see how it looks in your own spreadsheet/database program. Imported records can be found in job log 12<user>.job. Found duplicates is found in log file 13<user>.job.

Transport files containing data from several tables can be imported by selecting more than one table for import. When importing data, always select if duplicates should be dropped or replace existing records. Please note that importing large transport files can a take long, up to several hours in extreme cases with millions of records.

Printing Log files, menu File-Logfile

Use this selection to view or print program job log files (or any other text file you choose, for that matter). Job logs are normally named cr<user>.JOB where c/r stands for the menu column/row that created the job log. They can also be named after a certain user-written program like <program-name>.JOB. The log files $$<user>.JOB and $@<user>.JOB lists what different user is, and was, working with during the current and the previous database session.

Several files can be selected if you wish to view them all in a database Window. You can send a file to the printer (just the latest one). Chose cancel when you are done.

Editing text files in a text editor, menu file-Textedit

This menu item makes a call to a text editor, like the MS-DOS editor (version 5 and later). Upon installation a call is made to this program via the FD15.BAT file that calls ”EDIT Untitled”.

Printing text files and tables, menu File-Print

Here you can print **text files** or table content/status. For text files, you can also print a directory listing of the printed files. This may be useful when wild cards is used in text files path and name. Look for files with the BROWSE button. In order to print a group of files, you can then adjust a selected file name for wild card characters. When printing table content/status, first select the table. For printing the content you must enter starting (minimum) and ending (maximum) values for the tables first (normal) index (if it has several it is always the first one given in *Appendix A*.

Free table update, menu File-Update

Here it is possible to freely update records in any table. You must take help from *Appendix A*. Text field values are often selected by using the LIKE operator. The function of LIKE is the same as was described in *chapter 8* for data retrieval. Use the **Filter** button to define what records that should be affected. Use the **Update** button to define what should be changed. You can enter several updating statements, separated by colons (:) like this:

TextFieldX = ”string-text” : NumericFieldY= value : DatefieldZ = ”dd-mm-yyyy”

Besides constants, you can use data from other fields and/or use the standard mathematical functions +-*/).

Free table record delete, menu File-Delete

This part is used to freely delete records from any table. You first select a table. Then you should set a Display group filter. This is used to specify the restrictions on what records should be deleted. See also Filter button in previous section (updating records) and the table record details in *Appendix A*.

When you click OK, the database will tell you the total number of records in that table and how many that will be deleted. If you confirm the delete, the records will be removed. **Note:** Be careful in your work. You are advised to have safe copies of your database tables. Once you have removed the records, they cannot be restored.

Reorganizing tables, menu File-Reorgan

This menu item is used when you wish to delete duplicate records in a table, or for reorganizing an entire table. You may have to reorganize a table if an index file for that table has been damaged (a table consists of 3 or more files: the data file (.SBF file), the definition file (.SBD file) and one or more index files. Index files have file extensions consisting of a number referring to the field number of the field which is indexed. **Note:** Before reorganizing, it is recommended to print a file status (menu File-Print) and to safe-copy the database.

Change to other Superbase Programs, menu File-ChangePgm

This menu choice allows you to run another Superbase (.SBP) program. Such programs may be included in specific language versions of the database. You can also use it to go to another application of your own. If you choose the Start.sbp program, you will restart this program. If there are special programs for your language, you will find descriptions in *Appendix D*.

APPENDIX A TABLE FIELD DEFINITIONS

This appendix contains list of the fields in 16 of the 18 database tables. There is also a system (FDSYSTEM.SBF) and a help text (FDHELP.SBF) table. You may need these definitions for importing data from other systems or for making free updating or deleting of table records. Listed are table names, menu names through which the tables are accessed, each table's field names and indexes. The field data types means:

TXT	Text field, maximum no. of characters, U=Upper case characters stored
DAT	Date field, format 0d-mm-yyyy, for example 07-11-1997 (Nov. 7, 1997)
NUM	Numeric (floating/real) field. Maximum integer, decimal digits. "-" allows for negative values.
NMI | NML	Integers | Large Integers. Maximum no. of digits. "-" allows for negative numbers.
EXT	Text field containing a reference to an image or sound file name.
(**) | (*)	The field is indexed | is a part of a calculated sorting index

When preparing files for import, the data columns should come in the same order as the table fields. For text files, you should have the field separator character between each field - often ; or tab (ASCII 59 or 9). It is important that the texts in a text file don't contain the chosen field separator character. If so, choose another one and temporarily set the database to that field separator while importing the file.

The wrong decimal character (normally you use a decimal point) may cause the decimals to be dropped on import, or cause problems for other programs to which data is exported. **Note: Never use the same character as decimal and field separator!** The used date format in an import file must be the same as in the table. If it is not, it may be possible to adjust the database temporarily by changing its language.

Table FD21 **Menu: Series-Design** **Files: fd21uk.sbf, fd21uk.sbd, fd21uk.1**

SERIES	(**)	TXT	12U
TTYPE		TXT	6U
STITLE		TXT	60
SRESPONS		TXT	60
TRMNTMASK		NML	5
PRINTMASK		TXT	5U
OUTPUTFORM		TXT	12U
PAR1		NMI	2
PAR2		NMI	2
PAR3		NMI	2
PAR4		NMI	2
PAR5		NMI	2
SPAR1		NMI	2
SPAR2		TXT	8U
SPAR3		TXT	120
SNOTE		TXT	120
PMFILE		TXT	80

Trial series design, no. of treatment factors, levels etc.

Table FD22 **Menu: Series-Treatmnts** **Files: fd22uk.sbf, fd22uk.sbd, fd22uk.7**

TSERIES	(*)	TXT	12U
TNO	(*)	NMI	4
TCODE		NML	8
FCODE		TXT	8
TOPT		TXT	6U
TTEXT		TXT	100

List of treatments, laid out in field as well as factor averages and statistics report lines

Table FD23 **Menu: Series-Spec.trmt** **Files: fd23uk.sbf, fd23uk.sbd, fd23uk.7**

SSERIES	(*)	TXT	12U
SNO	(*)	NMI	4
TFAC	(*)	NMI	4
TLEVEL		NUM	-6.4
TUNIT		TXT	10
STEXT		TXT	100

Detailed specifications on individual treatment factors

Table FD24 **Menu: Series-Num.list** **Files: fd24uk.sbf, fd24uk.sbd, fd24uk.4, fd24uk.5**

CATEGORY	(*)	TXT	3U
NCODE	(*)	NML	8
NTEXT	(*)	TXT	36

Text library for standard treatment texts, varieties etc.
It has 2 indexes: Catgory+Ncode and Category+Ntext

Table FD25 **Menu: Series-Var.codes** **Files: fd25uk.sbf, fd25uk.sbd, fd25uk.4, fd25uk.5**

VCATEGORY	(*)	TXT	12U
VARIABLE	(*)	TXT	12U
VARTEXT	(*)	TXT	60

Library of standardized variable codes (elements)

Table FD31 — Menu: Trials-General — Files: fd31uk.sbf, fd31uk.sbd, fd31uk.1 – fd31uk.5

ID	(**)	TXT	12U
ISERIES	(**)	TXT	12U
PLACE	(**)	TXT	12U
PERIOD	(**)	TXT	12U
CROP	(**)	TXT	24
DISTRICT1		TXT	4
DISTRICT2		TXT	4
IRESPONS		TXT	60
PAR0		NMI	2
HARVESTS		NMI	2
ROTYEAR		TXT	4
NOTE1		TXT	120
NOTE2		TXT	120
NOTE3		TXT	120
NOTE4		TXT	120
NOTE5		TXT	120

Main table of the database. Establishes the relations between the individual trial and its series design and location

Table FD32 — Menu: Trials-Places — Filenames: fd32uk.sbf, fd32uk.sbd, fd32uk.1

PLACECODE	(**)	TXT	12U
HOST		TXT	60
HOSTADDRESS		TXT	60
SOILCODE		TXT	15
SOILTEXT		TXT	60
ALTITUDE		NMI	-8,4
LATITUDE		NML	-8,4
LONGITUDE		NML	-8,4
MAPFILE		EXT	80

Information about the trial locations

Table FD33 — Menu: Trials-Trialinfo — Filenames: fd33uk.sbf, fd33uk.sbd, fd33uk.7

IID	(*)	TXT	12U
ICODE	(*)	TXT	12U
IDATE		DAT	0d-mm-yyyy
IDC		TXT	4
ILEVEL		NUM	-6,4
ITEXT		TXT	30

Trial background information of all kinds - planting, fertilizing, pest control, harvesting time etc.

Table FD35 — Menu: Trials-Distrubut — Filenames: fd35.sbf, fd35.sbd, fd35.6

DID	(*)	TXT	12U
PNO	(*)	NMI	4
TRNO	(*)	NMI	4
GRP		NMI	4

The treatment distribution on the trial plots

Table FD46 — Menu: Printouts-Printlib — Filenames; fd46uk.sbf, fd46uk.sbd, fd46uk.5

LISTCAT	(*)	TXT	12U
LROW	(*)	TXT	4U
LTYPE		TXT	1U
LTEXT		TXT	88

General text library for printing field cards, labels, analysis forms, diaries, work lists etc.

Table FD64 — Menu: Outputs-Formulas — Filenames: fd64uk.sbf, fd64uk.sbd, fd64uk.10

CSERIES	(*)	TXT	12U
CROW	(*)	NMI	4
CALCODE		TXT	1U
VAROUT		TXT	16
VAR1		TXT	16
VAR2		TXT	16
FACT1		TXT	8U
FACT2		TXT	8U
CTEXT		TXT	30

Formulas for calculation of stored data in FD72 and FD73

Table FD65 — Menu: Outputs-Profiles — Filenames: fd65uk.sbf, fd65uk.sbd, fd65uk.11

RSERIES	(*)	TXT	12U
RCOL	(*)	NMI	4
ROPT		TXT	2U
RCODE		TXT	16
DEC		TXT	2U
TROW1		TXT	8
TROW2		TXT	8
TROW3		TXT	8
TROW4		TXT	8
RECALC		TXT	12

Result profiles of variables or other data for printing results and making statistics files

Table FD73 Menu: Base-Var.list Filenames: fd73sbf, fd73sbd, fd738

VID	(*)	TXT	12U
VVAR	(*)	TXT	12U
VPAR1		TXT	1U
VPAR2		TXT	1U
VDATE		DAT	0d-mm-yyyy
VAREA		NUM	7,4
VDC		TXT	4U
VSEL		NMI	-4

Keeps a list, with some extra information, of variables stored in table FD74, FD75 and FD76

Table FD74 Menu: Base-Averages Filenames: fd74.sbf, fd74.sbd, fd74.5

TID	(*)	TXT	12U
TVAR	(*)	TXT	12U
TRMN	(*)	NMI	4
TVALUE		NUM	-6,4

Stores all registered variables with treatment average data

Table FD75 Menu: Base-Plotvalue Filenames: fd75.sbf, fd75.sbd, fd75.5

PID	(*)	TXT	12U
PVAR	(*)	TXT	12U
PLOT	(*)	NMI	4
PVALUE		NUM	-6,4

Stores all registered variables with plot values

Table FD76 Menu: Base-Stat.pars Filenames: fd76.sbf, fd76.sbd, fd76.5

SID	(*)	TXT	12U
SVAR	(*)	TXT	12U
SROW	(*)	NMI	4
SVALUE		TXT	8

Table for storing statistical parameters from made analysis

APPENDIX B SPREADSHEET AND STATISTICS FILES' FORMATS

Importing spreadsheet data

You can, of course, always arrange your data according to the corresponding table in *Appendix A*. This is the way data is stored in this database - in deep and narrow tables (which are related to each other to form spreadsheet-like files when data is retrieved).

However, you might have trial data in columns on spreadsheets (Excel or Lotus 1-2-3). You can rearrange these data in the way described below and then you can import them to the database. Maybe you can do the same for data from "flat" database tables. The word flat means that the different registered variables are stored in pre-defined columns (as in a spreadsheet or in your statistical program's files).

When you import data spreadsheets or other flat database files as described below, you must store your data in files with extension.XLS (Excel) or .WKS/.WK3 (Lotus 1-2-3). You can only import the first sheet of a workbook and might not handle the latest spreadsheet versions. Save the data you wish to import on the top sheet and/or in an older file format. The spreadsheet file must be closed before import can be done.

Since the spreadsheet columns should contain also text information, normally use "general" or "text" column formats. You are advised to check these things out by preparing and importing a simple test sheet. Another restriction. **Note: Rows 1-4 should be text in order to ensure correct variable information. Integer and decimal numbers can be entered preceded by ' in order to ensure text format.** The example below shows a spreadsheet with some data. Columns B-F represents different registered variables. Spreadsheet rows 1-4 are header rows. Rows 5 and down contains the actual data, where plot or treatment average numbers is given by column A.

Column A is used for general information and for plot/treatment number references. The first column A row must contain the trial identification. Optionally, row 2-4 may contain series ID, location ID and crop. Plot/treatment numbers starts on row 5. After these numbers other text (separated by space from the numbers) can be entered. You can also have rows without plot/treatment numbers with any text but with no digits 0-9 (like Replicate I …. Replicate IV).

Columns B and right: Row 1 should begin with P,X,S, F or T directly followed by a registration date (if any). See allowed date formats in the example. P refers to plot values in plot number order. Treatment averages are labeled X if they are in the treatment list order or F if the follow the allocation in the first replicate. For text data the S or T labels is used in the same way as X or F above. **Row 2** stores the variable codes (be sure that it is entered as text information). **Row 3** is used for entering optional area (for later re-calculation of figures per m2 or hectare). Put 'sqm' after entered areas. **Row 4** can be left blank or used for entering decimal growth stages for the registrations, when appropriate.

On row 5 and down in column B and right the data values are entered for the plot/treatment numbers given in column A. If you have missing data just leave the cells empty. It should not contain a zero because then 0 will be stored in the database (which is not the same as a missing data).

Spreadsheet example.

	A	B	C	D	E	F	G	.....
1	071326	P16-08-95	P16-08-1995	X10091995	X	P11-09-95	P10-05-95	
2	R7-500	Y.RD	STS.SH	CNS	VW	MC	PL.SS.X	
3	C-28-1996	20.6sqm					0.5sqm	
4	Winter Wheat		15				25	
5		1	2	3	4	5	6	
6	Block I							
7	1 1A	12.34	100	98.3	765	17.3	210	
8	2 1B		95	96.8	732	19.3	184	
9	3 1C	11.7	98	97.3	743	18.8	196	
….								

Exchange of data from/to statistical packages

The statistical analysis of your field trial data cannot be done by the database program. The reason is simply that specialized statistical packages are more flexible and powerful than any built-in utilities in this database program. However, this means that you must be able to take out the data you wish to analyze, and to store statistical results, in the database.

You have two possibilities to extract data:
1. Export data directly from selected tables and the rearrange data in your statistical program.
2. Take out statistics files via the Base-Retrieve or Outputs-Stat.file menus as described in *chapter 7 and 8*

The second alternative is probably the most convenient. The various options are discussed below. In general, the **statistics files are flat files**. This means that all extracted data is tabled in columns. The data rows normally comes in plot order. Treatment averages will be assigned to the first replicate and to those plot numbers that have each treatment combination. The database program does this with the help of the treatment distribution table (FD35). From the retrieval menu you can alternatively take out a statistics file with only averages. In this case the data comes in the order of the treatment list (table FD22).

In *chapter 7* we described the different options for taking out statistics files. There you can choose between creating one file/trial or a common file, you can select to take out all variables or to use a result profile. You have always to prepare an use a profile in order to use the @ function -*described in chapter 3*-, to extract other data than factor levels, replicate number and registered variables. The Outputs-Stat.file menu also allows you to set different options for created text files.

If you use the more powerful selection possibilities in the Base-Retrieval menu, your options for the statistics file are limited. You can take out plot values or treatment averages, but you must always specify and use a result profile. If you take out data to text files, they will always be of the field-separated type where the separator character is the on set in the File-Setup menu.

If you wish to **pass on** some **parameters** to the statistics file, you can do this in table FD21, menu Series-Design. If you enter something in the ”Statistics code” and/or ”Statistic model” fields this information will come on one/two lines first in the statistics file - in the above order. These fields can also contain @ functions. For example, @P would fetch the number of replicates, @P6 the number of treatment factor combinations, @A04 the trial identification and so on.

Treatment distribution files

From the menu Trials-Fetchdist, special treatment distribution text files can be output. These function works only with trial identifications not longer than 8 character. The text files have a pre-defined, special format. The distributions for the trials are printed after each other in a sequential file named after the job log file, 36<user>.JOB. Each trial have the following rows:

Row	Content
1	Parameter row: Column 1-8: Trial-identification, Column 9-16: Extraction date in format ”ddmmyy” Column 17-20: 3 (= information items) , Column 29-32: no. of plots (=treatment codes)
2	General information row, items in 12 chr. columns (most of them blank): Column 1-12: Series code Column 25-36: Location code, Column 49-60: Crop (only left 12 characters)
3	Treatment code rows: Codes in plot number order in 6 character columns Each row can hold 20 codes

Transport files

From the database Base-Retrieve menu you can take out selected trials in a **transport file.** This file contains data from all the database tables where the selected trials identification, series or location code is found in table indexes.

From the database transport files are normally taken out in scientific format. A special description of the transport file format is included with this installation (as a MS Word file). The scientific format eliminates problems with different used date formats and decimal signs. Real values are exported with a 10 digit accuracy.

APPENDIX C SYSTEM FILES AND DETAILS

1. Description file and help texts

The description file is the most important part. The database uses them to load all language- and application-dependent texts. It is also used to generate all data-input forms the first time a new language or application is created. The description file should have the file extension .SB0. The name should be either a language or an application (like Francais.sb0 or breeding.sb0). The file has sections marked [section text] like the Win.ini file. These sections are sometimes used by the program and should not be changed. The rows beginning from the [Form Layout] section is only used when generating input forms the first time.

Help texts can be edited from the system setup menu, button SYSADM F1. Only users with ”sysadm” rights can do this. Help texts should be stored for the menu column 1-7. Each column can, for example, have a help page for each menu row. Structure the help text as the menu. The help text for menu column 2 can, for example, have pages from 20 and up to 29. Page 20 could be used as a chapter summary.

3. Calls to MS-DOS programs

In menu column/row 15,51,52 and 53 calls can be made to other Windows or MS-DOS programs. The ”15” call is supposed to call a text editor and the ”5x” may call raw data handling programs. All calls except ”15” is made with 9 command parameters that can optionally be used (blank parameter is printed as *):

1 Call comes from menu: 1...3 = menu ”51”...”53”
2 The user’s general work directory
3 Drive:\path for a directory where raw data should be stored
4 Drive:\path for storing a copy of raw data
5 Drive:\path for a directory to hold linked (merged) field data
6 Drive:\path for a directory with statistics results
7 Serial communications device and speed in text format like ”COM1:9600”’
8 Printer port for the printer used by the database in text format like ”LPT1:”
9 Username, the first 6 characters (if not known – Superb).

When the database starts, any Superbase programs with names FD*.SBP is also loaded (for example the FDSTAT.SBP program containing subroutines for exchanging data with statistics packages SAS and NCSS).
A program call to, for example, menu 5 row 1 (receive field data) is made in the following order (to programs in the database directory):
1 To a loaded subroutine within a loaded Superbase program
2 To another Superbase program called FD51.SBP
3 To another Windows or MS-DOS program *in the database directory* called FD51.EXE or FD51.COM
4 To a Windows 3.x program information file *in the database directory* called FD51.PIF
5 To a MS-DOS batch file called FD51.BAT.

4. Handling database users

The database can be run under all Windows operating systems except Windows CE. It can be installed on a local hard disk or on a mounted network disk and run either as single-user or a network version (with tables opened in shared mode. To change from single-user to network mode, you need a special program that modifies the database system table. This program can be obtained when purchasing the database program or any time later.

For network versions, an additional program, Netadmin.exe, can be started to monitor the use of the database of different users. A special network control file, Superbas.net, is installed. Special locking files (filename extension .SB!) keep track of which records in each table are used (locked) by which user.

Installed files: The files are installed in a tree structure under the selected installation directory. The only exception is that the SB30.INI file that must be placed in the Windows or Winnt directories. If the database is installed on a mounted drive on a network server, the server's safety backup routines may be used. In this case the database can only be accessed when the network is running. Another problem is that all programs, program extensions, filters etc. has to be loaded over the network before the database can start. This might take too long time to be convenient if is has to be done via telephone modem lines.

Database in network with local program files: You can install the database programs locally but keep the database on the network. In this case the network administrator should first install the database and start it the first time. When the database program starts, a language must be selected after which an empty database will be generated.

The network administrator should also change the database setup under its File,Setup menu. A printer and suitable printer fonts should be selected. With the SYSADM button other users can be added. See *chapter 9*. Even if the database is installed on a server, each user may have a local work directory (selected by each user in the setup). For best operation in a network, the database path should be mounted as a drive letter and network printers preferably be mounted on a port (like LPT2:). The users can be connected in 2 ways:

1. One way is to copy only the SB30.INI file to each user's Windows directory and to add a shortcut to the SBRTS30.EXE file in the database network directory. With this installation type, all programs etc. must be loaded over the network.

2. An alternative is to start Superbase locally and then switch to the network directory to start the database program. Then install the database runtime system locally on each user's computer. The runtime system will then start quicker and then switch over to the database network directory where the application will start (the start.sbp program is run).

Some .DLL files might also be required for Windows 3.x systems. These files are *Compobj.dll, Ctl3dv2.dll, Ole2conv.dll, Ole2disp.dll, Ole2nls.dll, Ole2prox.dll, Storage.dll and Typelib.dll.* In case any of these files are missing, they installation program will install them in the system directory. *If the files exist, they will not be replaced.*

The system table (FDSYSTEM) stores one record of settings for each user. When that user starts the database, a copy of his settings is fetched to a user-created work table. Each user is named after the network name of the computer/workstation. If no name exists, the user will be called "Superbase".

The system table uses the 40 first characters of a user name. Furthermore, it uses only the first 6 characters as part in naming temporary working tables and files. This means that the first 6 characters must be different for users that should be able to open the database at the same time. When the program creates job log files or temporary tables for a user, it starts with two characters followed by maximum 6 characters from the user name. Examples: A job log for menu Outputs-Calculate will for user "Superbase" be named 61SUPERB.JOB.

The user name of the computer where the database is first installed will automatically get system administrators rights. The SYSADM button alternatives in the File-Setup menu, should not be changed while other users are using the database. If the system administrator must move to another computer, he should be sure to first grant access and sysadm rights to the new computer/user. New users can get the following rights: 1=SYSADM rights, 2=normal read/write access users, 3=read only users (cannot edit records).

If the computer name is changed without that the new user was given access to the database, you can reset the system table in the following way: 1-Close the database and make sure that noone else is using the database. 2-Remove the file "headings.sbd" in the database directory. 3-Restart the database as the user who should have system administrator's rights. The removal of Headings.sbd will appear to the program as if you were making a "first" installation. Choose language. The definition files will be copied. All data will be kept except system settings for the new system administrator.