

Welcome to K Database Magic

K Database Magic is a unique product. It's accommodating a simple admittance to datasets like other database handling products. Firstly it doesn't concentrate maintenance of datasets, but offer up users lot of built functions never a product before. Various procedures make it one and only system, what the producer always supported and improve product.

The project started in summer of 2004 - unique by plan - what serve out the users easy to work. The product is user-friendly, everybody easy to use, edit, insert data to datasets. Beside all of them the user save the changes to project files.

K Database Magic is a professional multifunction database handling system. Shortly it's proud of the most important functions:

Data manipulation between two datasets: Help user to import & export datas with macros and functions (merge datas).

Compare two dataset by user specified unique fields.

Multiple SQL Editors Script Executing (SQL Manager): SQL edit, history of executed scripts, processing, watching.

Powerful data managing tools: data manipulation, create, copy, print, import & export other databases, etc.

Now the program supported the following database types: InterBase, Ms Access, Ms Excel, MySql(ODBC), Ms Sql, Oracle(ODBC), Foxpro, Paradox, dBase, text driver(*.csv)

And some other useful features:

- + Searching datas by barcode scanner
- + Viewing / editing data in tables
- + Multiple database connections
- + Export data to four formats (Excel, clipboard, csv, xml files)
- + Design platform, customizable program preferences: set up directories, colors, fonts, icons, etc.
- + Multilanguage interface

Copyright information

Release 2.8.3.0 2007. September
Copyright © 2003-2008 Sandor Kovacs. All rights reserved.

ochregray@ochregray.com

Irinyi str. 28
H-6000 Kecskemét
Hungary

<http://www.ochregray.com>

K Database Magic is a registered trademark of Sandor Kovacs. It's protected by international copyright law.

Technical support

I hope, **K Database Magic** product find yours approval because of s urface and easy to use, what compete with other database desktop applications. The built in help satisfy all needs about the program.

Mail address:

Irinyi str. 28

H-6000 Kecskemét

Hungary

E-mail: contact@ochregray.com

K Database Magic
Copyright (C) 2003-2008 Sandor Kovacs
Web: <http://www.ochregray.com>
E-mail: ochregray@ochregray.com
Developer: Sandor Kovacs

All Rights Reserved

License Agreement

You should carefully read the following terms and conditions before using this software. Unless you have a different license agreement signed by Sandor Kovacs. Your use of this software indicates your acceptance of this license agreement and warranty.

This End-User License Agreement is a legal agreement between you (either an individual or a single entity) and developer (Sandor Kovacs) for the software product identified above, which includes computer software and may include associated media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this LICENSE AGREEMENT. If you do not agree to the terms of this LICENSE AGREEMENT, do not install or use the SOFTWARE PRODUCT. Please do NOT order a registration key, if you do not agree to this license agreement.

License conditions

The SOFTWARE PRODUCT is protected by copyright laws and international copyright agreements, as well as other intellectual property laws and agreements. No part of the software or the manual may be multiplied, disseminated or processed in any way without the written consent of Sandor Kovacs. Violations of these conditions will be prosecuted in every case.

The use of the software is done at your own risk. The manufacturer and developer accept no liability for any damages, either as direct or indirect consequence of the use or abuse of this product or software by the end - user. The SOFTWARE PRODUCT is being licensed and not sold.

SOFTWARE PRODUCT LICENSE

GRANT OF LICENSE. This LICENSE AGREEMENT grants you the following simple non-exclusive rights:

Applications Software

You may install and use one copy of the SOFTWARE PRODUCT, or any prior version for the same operating system, on a single computer. The primary user of the computer on which the SOFTWARE PRODUCT is installed may make a second copy for his or her exclusive use.

Storage/Network Use

You may also store or install a copy of the SOFTWARE PRODUCT on a storage device, such as a network server, used only to install or run the SOFTWARE PRODUCT on your other computers over an internal network; however, you must acquire and dedicate a license for each separate computer on which the SOFTWARE PRODUCT is installed or run from the storage device. A license for the SOFTWARE PRODUCT may not be shared or used concurrently on different computers.

DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS

SOFTWARE PRODUCT as Freeware Version

It is protected by copyright laws and international copyright agreements, as well as other intellectual property laws and agreements.

Limitations on Reverse Engineering, Decompilation, and Disassembly

You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

Rental

You may not rent, lease, or lend the SOFTWARE PRODUCT.

Software Transfer

You may not transfer the rights of this end-user LICENSE AGREEMENT.

Data Transmission

You may not transmit the SOFTWARE PRODUCT, or parts of it, without the prior permission of developer.

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OR MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. Because of the various hardware and software environments into which K Database Magic may be put, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

Initial steps

Which are the most important functions, that you take note excellent:

1) *More important task is completed making project*

Can you load data tables to begin a new project, we can determine details these settings (for example open mode, filter) and save project various name, whatever you want.

2) *„Data pump / merge data“ equal data base import, export (project)*

The File|New DB Imp-Exp project under menu item to be found

We can easy move datas for each data table to another: unambiguous like nonprofessional users and give complicated conditions (programming) like professional users. It comes in useful to everybody.

3) *Data tables content comparison (project)*

The File|New DB Compare Project under menu item to be found

We can compare two dataset to set key fields and comparing fields. Then you see the differences between tables one saved dataset.

4) *Database desktop (project)*

The File|New DB Listing Project under menu item to be found

You can maintain datasets. Our jobs make it easy many features, built in functions.

5) *SQL manager*

The File|Tool windows|SQL manager under menu item to be found

We can connect immediate to databases, open tables and save settings; all that is easier, faster like a general project.

6) *Print data tables / datasets*

The File|Tool windows|Fast report builder under menu item to be found

Every datasets, SQL is printed to screen, printer every time, everywhere with selected fieldnames too.

Minimum system requirements :

Before you download software, please check if your computer meets the minimum system requirements!

Microsoft® Windows 98/98SE, Windows ME, Windows 2000, Windows XP, Windows 2003 Server

- 200 MHz Pentium II processor or equivalent
- 64 MB RAM
- 30 MB hard disc space for install
- Optional: BDE install (Borland® Database Engine), thus the program need not use local BDE. With installed BDE version (5.11) easy to modify deep database settings.

Version information

Version 1.0.5.0 - 01. 2005

BUG FIXES:

- Fixed a bug in Database import, export project: conversion mode is not correct, just worked append function
- Fixed a bug in Paradox table pack function repaired
- The same databases (name) in two different projects to open it becomes to inactive.

Version 1.4.7.0 - 03. 2005

NEW FEATURES:

- Direct access DB2, INFORMIX, INTERBASE 6, MSSQL, ORACLE, SQL Server, SYBASE databases with BDE.
- Tool windows icons placed next to menu bar.
- Associate files whatever data tables extension. You can drag and drop files from windows explorer and double click to open associated file(s).
- Open BDE administrator under program.
- One built-in file browser as tool window help to you open datasets easy. The most uses data directories we can save, load, change as alias name.
- We can exchange source and destination datasets on project windows.
- New option items can find in Environment Options window:
 - + Automatic up-down open tool windows
 - + On close don't ask condition the program

BUG FIXES:

- With the processor time significant discharge of load wasn't the program equal the minimum system requirements. This problem was fixed, removed the component.
- I exchanged window which display directories in Environment Options -> Input, Output directory choose. Here hided the window and directory selection was bad.
- Caption bar of tool windows was exchanged (Not work exactly under Windows XP).
- In the main menu -> Close all menu item is correct. The superfluous background windows closed and memory released.

Version 2.8.1.0 - 08. 2005

NEW FEATURES:

- The mouse scroll works in database grids [0.01]
- In Database import, export project took out the splitters (on second tab, center panels) I put panels instead of it, you can use drag and drop techniques to resize controls. [0.01]
- Syntax highlighter Text Editor (SQL, Pascal) [0.2]
- Add Pascal script to fields in Database import, export project [0.3]
- List physical field structures in Database Info tool window (dbase, paradox) [0.1]
- SDF creator project like project window [0.7]
- In Database import, export project log errors in Rich Edit [0.02]

Version 2.8.3.0 - 09. 2007

NEW FEATURES:

- From today the program is FREEWARE [0.01]

What we desire to make...

Usually from the projects

In the program sharply the functions are it apart parting. These windows are in the main window as „child windows“. We can save the settings to a file to reload later . (The project open and save functions in the main menu we could find.)

The **K Database Magic** project file extension is .kproj, what we can associate to the program under **File|Preferences** menu item.

You may carry things, related with treatment out the most of data bases possible in these projects, the further developments also serve purpose.

„Data pump / merge data“ equal database import, export

Otherwise known as a simple and useful **data pump**. This is rather easy, useful. Perhaps one of programmer's dreams come true. It's end of static troubles, demanding care to create new database and move records. This problem is solved, in fact we save the project (in error case), later reloaded it and use once again.

It is realized the usual database import, export (also you can find in program), but it is bid higher to make exactly, wide functions:

In the course of between moving fields:

- to assign source field to destination field
 - to assign fix value to field

 - to write functions and procedures, to evaluate calculated fields (macro)
 - DLL files and inside defined parameters functions to assign destination field
 - We can add script written in Pascal to field
- We can test macro and DLL functions.

We can replace characters to another to use various code pages, but we can also create own character sets.

These entire can we create a log file. (These files we can find under the program's directory **TempLog** sub-directory.)

Suggestion: we are arranging the effective force strengths in chronological order, because it will be easier to find the file wanted like this.

Database content comparison

How many times we need compare two datasets records exactly.

Two source datasets must be equal key field order!

After data comparison the changes is saved a dataset. This is a comma separated text file, what we can use up later (the program know .csv format, it can handle as data table). These files we can find under the program's directory **TempCompareDBs** sub-directory.

Suggestion: we are arranging the effective force strengths in chronological order, because it will be easier to find the file wanted like this.

Database desktop, listed

It's sure, that you have be seen database desktop, viewer applications. Well, you can find it into K Database Magic like a project.

If you choose the dataset (besides finer setting - filter, exclusive use, to open index, etc.) you can:

- search, sort, filter, replace
- zap table, undelete deleted records (dBase)
- import, export datas
- rename, delete table
- repair index

Search functions:

- ascending / descending field
- special sorting
- search and replace field value
- go to record number
- set / drop filter, add / delete selected field value to set filter

Other useful functions:

- pack table (dBase, Paradox)
- undelete deleted records (dBase, Paradox)
- remove duplicated records (dBase, Paradox) to set the key fields
- refresh indexes, if the date of index file is older than dataset
- remove index flag from dBase file
- import (clipboard, csv) and export (excel, clipboard, csv, xml) dataset also selected records

SDF text file converter

If you have such a file in datas wasn't separated fix characters and you would like to move to dataset, this project solves your problem.

Requirements are: one row number of characters must be equal to next row number of characters (rows means record separated CR/LF)!

Specify schema text file, but the best choice is schema database file, because you don't fill position valuables, instead of this grab dataset structure and you can copy datas immediately to dataset.

How can be use

Main window of program

Menu structure

File menu

New DB Imp-Exp project

It's open a new project, which we can input/export datas between two datasets.

New DB Compare Project

It's open a new project, which we can compare datas between two datasets.

New DB Listing Project

It's open a new project, which we can edit a datasets.

Open Project...

It's open a new K Database Magic (*.kproj) project.

Reopen

It's reopen the last ten saved projects.

Save Project

It's save the projects.

Save As Project...

Preferences...

Let the **File|Preferences** choose menu item, which visualizes the **Environment options** dialogue window. You adjust necessary environmental preferences here.

Tool windows

Database info

Every time, everywhere can you see database information. It display fast and practical database index; field names, size, etc.

SQL manager

We can connect immediate to databases, open tables and save settings; all that is easier, faster like a general project.

Fast report builder

Every datasets, SQL is printed to screen, printer every time, everywhere with selected fieldnames too.

Barcode selector

We can find valuables with barcode scanner: set it to any field. It can set field value in database insert / update mode.

Alias replacement

If you set hardly dataset properties in project window s and you like receive another project, you can do this easy. Just use this tool window...

Database browser

You can see that K Database Magic v1.4.7 has got well developed association. Use this tool window to search database files on your drives and click on it and see detail datas. (You can save your browsed directory.)

Exit

Exit from the program.

Window menu

Windows list

We could find here all opened windows. Clicking on menu item we could activate that window.

Tile Horizontally

A window is tile horizontally

Tile Vertically

A window is tile vertically

Cascade

Arrange

Previous

Previous project window

Next

Next project window

Minimize All

All project windows minimize

Close

Close the current project window

Close All

Close all project windows

Help menu

Contents

Topic Search

Tip

What's this?

Version info...

Version information

About

Other functions

Main status bar below:

- Click on status bar "small book" to view help.
- Click double "num, scroll and caps lock" to on or off it.
- View the date lower right.

Project windows

Database import, export (merge data) project

Preferences, which must do on tabs:

1. page tab:

We can choose source and destination datasets in database choosing dialogue , after that which properties can find the window left side (up and down).

On the right can we see the dataset field's names. The key fields (unique) – if the dataset has primary index – select on right panel, but you might select or unselect fields: Continuous push the CTRL key and push the mouse left button on field name.

2. page tab:

You can see on the left destination datasets properties, on center the source dataset fields and valuables, on the right both the two dataset mapping result.

1.) You are keeping mind the fields of destination dataset properties and you can adjust the conversion mode. You can be data:

- *add*: automatically, stepping from record to record takes up the datas . It's not search on dataset primary key. This is the fastest conversation function .

- *add and update* (need the primary keys)

- *update* (need the primary keys): It copies just records, that valuables equal in the primary keys valuables.

- finding records to *delete* (need the primary keys)

Click on "move record" combo box to move records from source table to destination . These are deleting from source.

You also add here so that only one of certain part (interval of the database) copy, move or delete : the value must add the start record number and record count .

2.) Proceeding upside-down:

- we can order the source field to destination field. Please select field (in source and destination too) with mouse. The selected item (fields) to push down "arrow" icon and you see on right side result table.

- we can order fixed value to the dataset field

- we can write functions and procedures. The result value will be adding the destination field. We can calculate to evaluate valuables. If we made the function, we save it. (see details on reference)

- we can attach DLL file function with special parameter. You can use anything what you need in the DLL file. The result value can we add, what you add to a destination field. (see details on reference)

We can also test the macro and DLL function (see the Dialogs chapter part)

3.) We can find data grid in window part the lower right , which the logical value is making fitted. We must supply the valuables: if the dataset's field type is not logical value , which "logical name" attaché to the field. Example: if the source field data type is character , which logical value replaces it? Let it be "FALse" character string equal false logical value and "TRue" character string equal true value.

Which contains the result data grid columns (from left to right) :

- Show the conversion mode: field equal, value, function, etc. (see above)
- Source value, which contains other value depending on conversion mode .
- Destination field name
- Destination field type
- The source and destination fields conversion type *
- To change characters / letters on field valuables (see below)

If you click on the column of data grid, then you immediately see the conversions and edit on left side of window.

* The user can define / could force it the conversation type . He uses the following keys:

- „**b**(oolean)“: logical type

- „c(urrency)“: currency type
- „d(atetime)“: datetime type
- „f(float)“: float type
- „i(nTEGER)“: integer type
- „s(tring)“: string type
- „t(ext)“: text type

We have possibility to ask a possibility with converse , which calculated from the machine which feels respect for organization of a converse between the dissimilar type of fields.

3. page tab:

We have possibility to change characters with another one. We can use defined forward valuables (From ANSI to DOS852 or. From DOS852 to ANSI), but you can append new valuables here, save it, rename, delete; you maintenance (see the menu).

Select the conversion name (combo box see the window's right side) and you can attach it to one conversion item or all each other (see the menu).

4. page tab:

If you start the conversation method, the program writes a log continuously from conversation or another errors, which we can save. You can find error files in program directory TEMPLOG subdirectory. The saved log file type might be Rich Text or simple text file.

You stop conversation anytime.

An indicator points out the process which is stretching out comfortable showing by .

Happen extra possibility:

- Dataset „clone“: The source dataset will display on destination database . The two dataset will agree after the bringing about structurally (fields and indices) .

Database content comparison project

Preferences, which must do on tabs:

1. page tab:

We can choose source and destination datasets in database choosing dialogue, after that which properties can find the window left side (up and down).

On the right can we see the dataset fields names. The key fields (unique) – if the dataset has primary index – select on right panel, but you might select or unselect fields: Continuous push the CTRL key and push the mouse left button on field name.

2. page tab:

Here you associate the key fields and comparison. You must click on and select source and destination field name. Then you click “add key field” or “add compare field” icon on bottom to stand the fields into pair. You must be **ordered table** to compare good!

The program searches on destination dataset by key fields. Compare compared fields to show you if not equal the value.

3. page tab:

Here you find the source, destination and compared result tables. You can be continuously the comparison, break/cancel it (the comparing method proceed from first to last record) or step it forward and backward.

An indicator points out the process which is stretching out comfortable showing by .

Database desktop, listed project

Preferences, which must do on tabs:

1. page tab:

We can choose source and destination datasets in database choosing dialogue, after that which properties can find the window left side (up and down).

On the right can we see the dataset fields names. The key fields (unique) – if the dataset has primary index – select on right panel, but you might select or unselect fields: Continuous push the CTRL key and push the mouse left button on field name.

2. page tab:

It's sure, that you have been seen database desktop, viewer applications. Well, you can find it into K Database Magic like a project. If you choose the dataset (besides finer setting - filter, exclusive use, to open index, etc.) you can:

Search functions:

- ascending / descending field
- special sorting (see **ABC order dialogue** window)
- search and replace field value (see **database record search and replace** dialogue window)
- go to record number
- set / drop filter, add / delete selected field value to set filter (see **database filter** dialogue window)

Other useful functions (proceeding upside-down):

- pack table (dBase, Paradox)
- undelete deleted records (dBase, Paradox)
- undelete records (dBase, Paradox)
- remove duplicated records (dBase, Paradox) to set the key fields
- refresh indexes, if the date of index file is older than dataset
- remove index flag from dBase file
- rename, delete table
- import (clipboard, csv) and export (excel, clipboard, csv, xml) dataset also selected records

Tool windows

Database info

Choose the **File|Tool windows|Database info** menu item to show the **Database info** dialogue window. You get information about the selected dataset (top window combo box). (If you see project window, than you choose two dataset – source and destination – otherwise one.)

Select a window with the current window: Choose a child window (at variance with this window), then click on the **Database info** window again.

We could see dataset's functional pieces of information in upper data grid:

- table name
- dataset directory
- filter, opening conditions
- etc.

On left from dataset prepared statistics:

- field count
- record count
- indices count

On center the field and index properties:

- fields name, type, size
- indices name, names of open fields indexed, primary, own, etc.

On right could you see table's hierarchy like general Ansi SQL in form.

SQL manager

We could easily manage our files here, SQL commands we can run and the datasets used we could save, we could reload.

The more important functions :

Database open

We could open any dataset to be found with the help of the **Dataset selector**, where just we must add the directory name.

On the left side we can find the listed datasets. The selected table's contents show in right side data grid, properties show bottom side data grid.

If you use the **User SQL tables** mode, then you could see the saved user SQL datasets on left side list box. The selected table's contents show in right side data grid, properties show bottom side data grid.

Save as new

A select dataset can be saved to user SQL Tables (User SQL tables) on new name.

Save

A select dataset can be saved to user SQL Tables (User SQL tables).

Delete

A select dataset can be deleted from user SQL Tables (only **User SQL tables**).

Choose field value

If you are in table's contents showing data grid and select a field and choose this function, then you could insert or edit to last selected dataset's field. (it serves setting off the clipboard.)

Choose a child window (at variance with this window) and into this click on a data grid, then go to the **SQL manager** and use this menu item.

Remarking: the destination dataset must be for editable!

Follow database

Follow dataset records with SQL table to set fields. The data editing function is easing really, what we position from record to record in a selected data grid with record positioning in **SQL manager** data grid.

That why also important this, behold it practical net income : It happens, that we want to filter with such a condition and index according to fields, what it's complicating our task to set properties of dataset filter and create new indices. That we could easily carry this out with write a SQL script, not speaking also about task, which datas receive from more datasets. SQL table can we connect with editable dataset and with conditions we could modify records like this.

So write our SQL script in **SQL manager** and let us save with the above-named method.

On **Options** page tab could we adjust the key fields to be using field name order by on right side. To use drag and drop method to take up fields list box named by **SQL following fields**, use **Del** key to delete from list.

We could also set short and long names of the dataset here.

On **Editor** page tab can we edit our SQL script.

Let are checking **Follow database** menu item in, choose another window and into this a data grid, then go to **SQL manager** and you see the result to move record step by step.

SQL file open

We could open any text file, which in can find SQL scripts. We could find the loaded text on **Editor** page tab.

SQL file save

We could save text to be found on **Editor** page tab.

SQL batch run

More SQL script running. Possible mistakes could you see on the **Editor** page tab under **SQL text** field.

Structure of more SQL script containing text file :

Comment: start row with "/" character

Separated character: which separates different SQL scripts apart from one another, use the ";" character into a new line.

Example:

```
//Delete all record from animals.dbf:
```

```
delete from animals.dbf
```

```
;
```

```
//Add new record:
```

```
insert into animals.dbf
```

```
(NAME, SIZE)
```

```
values ("PONTY", 10)
```

```
;
```

Execute SQL

Execute a SQL script. Use this usually if the script no result value (insert, update, delete, etc.).

Open SQL

Open and run a SQL script. Use this on "select" SQL script.

Previous command and Next command

It's historic looking backward the running or opened SQL directives. We could choose previous SQL script.

Fast report builder

In tool window

Any dataset list to screen and print.

Let' choose another window (at variance with this window) and into this a data grid, then click on the **Fast report builder**.

Top of this we could set the name of the list, we could select listed fields under this :

- with keyboard: in activated data grid push ENTER key.

- with mouse: check on it.

We could turn inverse selection with bottom button.

The print button print just selected fields.

Page preview

Here you print the report, to save report to later printing.

Printed report save norm formats:

- Quickreport format (.qrp)

- HTML

- Text (.txt)

- Comma separated (.csv)

The normal **print** function to print report direct to printer, and with **print...** menu item first we could set the printer and parameters then we can print.

Structure:

1. Upper toolbar:

Upper menu item in toolbar. You could set page size (%), page number, etc. in edit boxes.

2. Page viewer:

The page viewer shows you the report one page. We can set in menu the view: 100%, page fit. The page viewer scrolls the page with scroll bar.

But you may scroll it variously also:

with keyboard:

Click on this with the mouse, then

'W' - Up (Automatically go to previous page, if you go to top of page.)

'S' - Down (Automatically go to next page, if you go to bottom of page.)

'A' - Right

'D' - left

with mouse:

Push the mouse left button and move any direction.

3. Bottom side status bar:

In the lower left corner in percent visible displays scale all pages and prepared pages. Next to it finding text contains name and page number of the printed list.

Barcode selector

In a selected dataset can we search datas with barcode scanner. This tool window is like other tool windows its working essentially.

Let' choose another window (at variance with this window) and into this a data grid, then click on the **Barcode selector** window again.

In the upper combo box can we select the dataset, which could we block (fix dataset) it to be found on its opposite prompter with a "padlock" icon. The list element implying the name will stand still onto effect of freezing, automatically not the list box rolling down freshly.

We still find in addition to this on the left side a - it contains field names of dataset - list box. We can move field names to list box on the right side with buttons on arrow icon. It may list more field name here. This is important in such an occurrence, if value of more fields is equal value of barcode. You can change search order with up and down icon on button.

The button named **Select row** selects one row in dataset.

You can see list of read barcodes in memo box on bottom, which you can save or delete value with buttons on right side.

You can set com ports with the check button named **Options** to use barcode scanner. You can use on panel on left side you set:

You can set to connect a barcode scanner to computer.

Port

Use Port property to set serial port number.

Baud rate

Sets character transmission speed. Baud Rate property represents the speed at which characters are sent or received via RS232 connection.

Data bits

Sets the number of bits in the bytes transmitted and received.

Stop bits

Sets the number of stop bits per character.

Parity

Use Parity property to enable or disable parity checking, error replacement and parity type.

Flow control

Use Flow Control property to set flow control type to hardware, software or none.

It can search only complete utterance (barcode).

If it finds out automatically and select a data grid on selected window.

Or it selects one row in a data grid and you can not push button named **Select row**. (You can set these valuables on checkboxes)

You can save changes to push **OK** button, pushing **Cancel** button to reset changes.

Dialogs

Intro window

The most important information is here about the program and developer. We could close with button on upper left side.

Environment preferences

Preferences, which need do on:

1. Program set view

We can set colors, fonts with **schemes** in the combo box. We can set gradient color to click on **arrow** icon and set own colors to lighten or darken it with slider on right .

On the **background image** element we can select any picture to click left button mouse. Click on right button to view popup menu: use a background image, gradient color or not .

You can view the program's icons.

Comment: These resources are flexibly manageable extremely. We could replace the icons , we could decorate the employment with pictures , we can add new **schemes**, etc. I don't get embroiled in going into details. On the website to this you receive directive.

2. Folders, directories, user specific

We can select **input**, **output**, **background images** folders. It must contain program view (see above) and languages (see settings on homepage) directories in the input folder. Files is saved by **SQL manager** will display under **SQLManFiles** subdirectory and the **charchg.ksh** file, which contains characters conversation types on **database import/export project**.

Also you can set (example: user name is on the report upper left corner) user name, address and e-mail address.

3. Functional specific of program:

- show hints
- automatically save project on close
- the instead of configured, installed BDE we could force, so that the program uses version the enclosed BDE under program directory
- in the explorer click ***.kpj** extension to start the program
- we could choose the foreign language

Database chooser

We can choose currently among three kinds of datasets.

1. Delimited characters, text database file:

The used Ms. Excel's outcome in the program practically, simply we can handle it to visualize .

The **arrow** button helps us to browse the file and set which characters delimited fields . Here you must write one character, exception: **space**, **tab**, **esc**.

You can view field names if you click on mouse in text box named **Fields**. If you not see try it another character. You can delete, update and create field name . You can save new field structure to use **create file** button.

The fields must be under each other like list items !

2. BDE table

It handles the most notorious datasets in table: InterBase, Ms Access, Ms Excel, MySql(ODBC), Ms Sql, Oracle(ODBC), Foxpro, Paradox, dBase and text driver.

We could do following preferences:

- choose **alias** name or click on **arrow** button to browse database directory
- the dataset table name
- open index file to use it
- table type (**default** value)
- dataset table refresh mode (always, where the key change on primary index)
- automatically refresh
- open it exclusive, only one user uses the table
- filter
- the filter mode may be case sensitive or partial key
- delete all records in table

3. BDE SQL

Same as what I write in upper, but here you can write complicated **select** statements. (Other SQL statements you can use build-in **SQL manager**)

Because we can open here select statements, we could do following preferences:

- choose **alias** name or click on **arrow** button to browse database directory
- write in text box the SQL (**select**) statement
- dataset table refresh mode (always, where the key change on primary index)
- automatically refresh
- live **select** SQL statement (under developing)
- filter
- the filter mode may be case sensitive or partial key

We can save the changes to push the **OK** button, or cancel it **Cancel** button.

You can find useful information from BDE (**Borland Database Engine**) in **Reference** part.

Function/procedure test

You can test functions, procedures what you wrote macros and DLL functions in **database import, export (merge data) project** window. We could see field valuables of source and destination datasets on benchmark, or step one record forward, backward if you use button series on window below.

Parameters and valuables, result valuables into three groups its dividing:

1. Attach field valuables of source dataset on DLL function benchmark (parameter)

The input value is under **[Fields]** note to be found. (We could handle so it as a *.ini file.) Here you can see fields of source dataset and these valuables according to types. (for example: the string value border by " " symbols)

The result value will be result value of called DLL function.

You can see DLL function name and call procedure name on below.

2. Attach field valuables of source and target dataset as parameter on DLL function (parameter)

Here you could not find the input values (this is not different, as value of selected field of source dataset); the result value will be result value of called DLL function.

You can see DLL function name, call procedure name and associated field on below.

3. Run macro

The variables look as follows:

- field names of source dataset receive „**s**” key signature
- field names of destination dataset receive „**d**” key signature
- the **result** variable is the result value, what value appears to selected target field
- additional variables: the **date** variable show the actual date, the **now** variable show the date and time.
- in addition to this we can use and create own variables...

You can see written macro script on below.

Because of step by step moving dataset record change the valuables thus calculated valuables also change.

You can find useful information from macro and DLL functions in *Reference* part.

Database login window

You have to login into a database in some occurrence that you open tables (SQL servers, access). This time you must give your user name and password .

If we want that, let the window not appear (it don't show this window), then click on checkbox.

Login to database press **OK** button, or cancel it **Cancel** button.

Dataset filter window

Database filter is making quick and an easier selections possible .

You can find dataset **field names** on window side left below. Click on it double with mouse and view in edit box.

You can use the most frequent operators what you could find on **sets of buttons**.

If you ready the written filter condition you could validate the condition with button.

Example filter condition:

(NAME <> 'Urban') AND (PRICE >= 1500)

Search record and replace value window

You can search records and replace values in dataset with this window .

You can see field names of dataset in upper grid **first column**. Push down or up if you have to search one or multiple records. In upper grid **second column** you can set the search condition, in **third column** set the replace value.

Options:

- case sensitive
- whole words only

Scope:

- global, all records
- search in dataset only selected fields

Search, replace functions: (with buttons)

- go to the first find record
- previous, next record
- you can select the searched records
- replace the value
- replace all values with the condition

The last two occurrences must be the dataset in edit mode!

Sort records window

We could sort records according to multiple fields (max.: 5). This sorting is operating only on Paradox and Dbase datasets.

You can find dataset fields on left to move right side list box with arrow buttons. This is a sort field names , and push button below to change sorting direction. There is list box items „+” or „-” sign. (default is „+”)

We can set with buttons the sorting direction and case sensitive setting .

Click **OK** button to start the sorting.

(Attention: This is a physical sorting, therefore you have to reindex your dataset's indices)

Remove duplicated fields window

We could remove duplicated records in a dataset according to multiple fields (max.: 5). This sorting is operating only on Paradox and Dbase datasets .

The primary field – what you remove duplicated field - have to select in a project window to choose a list box. (see in *database desktop, listed project*)

This is a sort field names, and push button below to change sorting direction. There is list box items „+” or „ -” sign. (default is „+”)

We can set with buttons the sorting direction and case sensitive setting.

Click **OK** button to start the duplicated fields removing.

(Attention: This is a physical sorting, therefore you have to reindex your dataset's indices)

Reference

Use the datagrid object

You could view record in a grid rows (record) and columns (field names) . It helps you to view more information sorted list.

Structure:

- Row: It means records.
- Column: It means dataset field.

Columns and status bars.

- 1. columns: You could view the dataset fields .
 - 2. status bar: If you search or filter in grid you view it (read only mode).
 - 3. status bar: You could view dataset states. (for example: record number; dataset in append, edit mode etc.)
- (Some case only two page header perform)

Usable keys:

Navigation:

- arrow keys - move one cell left, right, up and down and select it.
 - PageUp - PageDown - turning pages up and down in grid .
 - Ctrl+Home - Ctrl+End - go to first and last record in dataset .
 - Home - End - go to first and last column in dataset .
- Enter - You can use a grid to modify existing data and enter new value. You can cancel all edits for a record by pressing **Esc** in any field before moving to another record.

Operations:

- Search in data grid - (only read only mode) Stand to column with arrow keys what we search on. Start to write characters. If it searches in dataset, the cursor goes to record. Certainly you could search multiple characters that write it continuous. You can use all keys, push **Backspace** if you wrong.
- Filter - (only read only mode) Hotkey: *Ctrl+S* Function of it: Filter in data grid. You can set filter condition upper 2. status bar, - or from-to interval (in such a case use '-' character to separate values!) – what you can see filtered records in data grid. You could finish it only with this hotkey (ctrl+s). You could filter this filtered dataset too.

Ctrl+S
Something value
Ctrl+S

or

Ctrl+S
something value-something value
Ctrl+S

You couldn't filter and search as data grid like as not see two statusbars above! (If you are in **inplace edit** mode, you can see if the field value starts to edit, then come out to press ENTER , then you must use Ctrl+S keys.)

Get menus with mouse:

If you click on grid with mouse right button, you can do:

- Insert: insert new record
- Delete: delete record
- Edit: edit record
- Goto record...: go to record number

- Find and replace...: call search and replace dialog
- Sort...: call sort order dialog
- Sort ascending: sort field ascending
- Sort descending: sort field descending
- Print: print data grid with report builder
- Import records: import records (clipboard, *.csv)
- Export records: export records (clipboard, *.csv, *.xls, *.xml)
- Refresh: refresh data grid values

Get menu: (optional)

- Old states: You could choose previous dataset states (search, filter).
- Table columns: You can set column visibility.

You can change column width and row height like MS Excel.
 You can drag and drop columns, change order.
 Double click on column header to sort order.

Relationship of BDE and the program, handling BDE

It needn't the operating system have got an installed BDE !

You can set dataset alias names in BDE Administrator, what you can find installed BDE application under "C:\Program Files\Common Files\Borland Shared\BDE\bdeadmin.exe" subdirectory (or in system console).
 You can find locate BDE in K Database Magic install directory under **BDE** subdirectory. You have to open IDAPI.CFG configuration file in BDE with Object|Open configuration... menu item. Then you can append alias names, configure local BDE settings.
 You can use in English BDE help. (in BDE program F1, directory: C:\Program Files\Common Files\Borland Shared\BDE\)

Use program local BDE:

- **File|Preferences** dialog, program uses version the enclosed BDE under program directory
- if the program terminated unexpectedly, restart the K Database Magic and close it normally. This needs for that, the BDE rewrite changes to registry. (All BDE application must be restarted for changes to take effect.)

Macro condition

Operators:

arithmetic:

addition, subtraction, multiplication, real division (x and y is an integer -type or real-type expression)
 $x + y, x - y, x * y, x / y, x ^ y$

relational:

equality, inequality, less-than, greater-than, less-than-or-equal-to, greater-than-or-equal-to (x and y is an integer-type or real-type expression; result is logical value)
 $x > y, x < y, x >= y, x <= y, x = y, x <> y$

string relational:

equality, inequality, less-than, greater-than, less-than-or-equal-to, greater-than-or-equal-to (s, t is string; result is logical value)
 $s > t, s < t, s >= t, s <= t, s = t, s <> t$

logical:

negation, conjunction, disjunction, exclusive disjunction (a, b is logical value)
 a AND b, a OR b, NOT(a)
 x in [...] (for example: 12 in [22, 12, 3] result=TRUE is logical value)

create variable:

x:=formula (or value)

free and clear variable:

create variable ("s" is variable name) s:=23
FreeVar(s);

Is live variable ("s" is variable name). If variable exists, the result is logical value:
ExistVar(s);

Type conversation:

Logic(x)

Converts an integer to a logical value.

Numeric(s)

Converts a string to integer.

String(x)

Converts an integer to a string.

Char(x)

Converts an integer to character.

Ascii(s)

Converts a character (ASCII) to integer.

Eval(formula)

This applies all types (see Clipper language) evaluate function, where *formula* is a string type formula. The result value is the evaluated formula. The formula is bordering [...] sign. (*Eval(s)*, where *s* is a string type variable)

function *NumBase(x,base)* : string

function *BaseNum(s,base)* : integer, ahol a base értéke <2..16>

Mathematical operations:

x Div y

Integer division: the value of $x \text{ div } y$ is the value of x/y rounded in the direction of zero to the nearest integer. (x and y is an integer-type or real-type expression)

x Mod y

Remainder: the mod operator returns the remainder obtained by dividing its operands. In other words, $x \text{ mod } y = x - (x \text{ div } y) * y$. (x and y is an integer-type or real-type expression)

functions:

Abs(x)

Abs returns the absolute value of the argument, X . X is an integer-type or real-type expression.

Frac(x)

Returns the fractional part of the argument X . X is a real-type expression. The result is the fractional part of X ; that is, $\text{Frac}(X) = X - \text{Int}(X)$.

Trunc(x)

Truncates a real-type value to an integer-type value. X is a real-type expression. Trunc returns an integer-type value that is the value of X rounded toward zero.

Sqrt(x)

X is a floating-point expression. The result is the square root of X .

Ln(x)

Ln returns the natural logarithm ($\text{Ln}(e) = 1$) of the real-type expression X .

Exp(x)

Exp returns the value of e raised to the power of X, where e is the base of the natural logarithms.

Cos(x)

Returns the cosine of the angle X. X is a real-type expression that represents an angle in radians.

CTg(x)

Returns the cotangent of the angle X. X is a real-type expression that represents an angle in radians.

Sin(x)

Returns the sine of the angle X. X is a real-type expression that represents an angle in radians.

Tg(x)

Returns the tangent of the angle X. X is a real-type expression that represents an angle in radians.

ArcSin(x)

Returns the inverse sine of X. X must be between -1 and 1. The return value will be in the range $[-\pi/2.. \pi/2]$, in radians.

ArcCos(x)

Returns the inverse cosine of X. X must be between -1 and 1. The return value is in the range $[0.. \pi]$, in radians.

ArcTg(x)

Returns the arctangent of X. X is a real-type expression that gives an angle in radians.

ArcCtg(x)

Returns the inverse cotangent of X. X is a real-type expression that gives an angle in radians.

MaxVal(x [,y, ...])

Returns the largest signed value in the Data array.

MinVal(x [,y, ...])

Returns the smallest signed value in the Data array.

SumVal(x [,y,...])

Returns the sum of all the values in the Data array parameter.

AvgVal(x [,y, ...])

Calculates the arithmetic average of all the values in the Data array parameter.

Character

operations:

concatenation: *s || t*

comparison: *s Like t*, where (%,_) it uses as substitute characters

s Wildcard t (*,?)

functions:

function *Length(s)* : integer

Returns the number of characters actually used in the string or the number of elements in the array.

function *Pos(t, s)* : integer

Pos searches for a substring, *t*, in a string, *s*. *t* and *s* are string-type expressions. Pos searches for *t* within *s* and returns an integer value that is the index of the first character of *t* within *s*. Pos is case-sensitive. If *t* is not found, Pos returns zero.

function *Trim(s)* :string

Removes leading and trailing spaces and control characters from the given string S.

function *TrimLeft(s)* :string

Returns a copy of the string S with leading spaces and control characters removed.

function *TrimRight(s)* :string
Returns a copy of the string S with trailing spaces and control characters removed .

function *Upper(s)* :string
Returns a string that is a copy of S, converted to upper case.

function *Lower(s)* :string
Returns a string that is a copy of S, converted to lower case.

function *Copy(s,x,[y])* :string
s is an expression of a string or dynamic-array type. *x* and *y* are integer-type expressions. Copy returns a substring or subarray containing *y* characters or elements starting at *s[x]*. The substring or subarray is a unique copy.

function *CopyTo(s,x,[y])* :string
s is an expression of a string or dynamic-array type. *x* and *y* are integer-type expressions. Copy returns a substring or subarray containing *y* characters or elements starting at *s[x]*. The result value is substring.

procedure *Delete(s,x,[y])*
Removes a substring of *y* characters from string *s* starting with *s[x]*. *s* is a string-type variable. *x* and *y* are integer-type expressions.

procedure *Insert(s,t,x)*
Merges *s* into *t* at the position *s[x]*.

function *Replace(s, t, v, bReplaceAll, bIgnoreCase)* :string
Replace in *s* search with *t* string and if find replace it with *v*.
bReplaceAll (if find replace it all) and *bIgnoreCase* (ignore case sensitive if 1). Their values are 1 or 0.

Date, time functions

function *Year(s)* : integer
Obtain the year represented by a specified TDateTime value.

function *Month(s)* : integer
Obtain the month of the year represented by a specified TDateTime value. Returns a value between 1 and 12.

function *Day(s)* : integer
Obtain the day of the month represented by a specified TDateTime value. Returns a value between 1 and 31.

function *WeekDay(s)* : integer
Obtain the day of the week represented by a specified TDateTime value. Returns a value between 1 and 7, where 1 indicates Monday and 7 indicates Sunday.

function *Hour(s)* : integer
Obtain the hour of the day represented by a specified TDateTime value. Returns a value between 0 and 23.

function *Minute(s)* : integer
Obtain the minute of the hour represented by a specified TDateTime value. Returns a value between 0 and 59.

function *Sec(s)* : integer
Obtain the second of the minute represented by a specified TDateTime value. Returns a value between 0 and 59.

function *StrToStamp(s)* : integer
Parse a string that specifies a date as integer type.

function *StampToStr(x)* : string
Parse a integer type that specifies a date to string.

function *StampToDateStr(x)* : string
Converts a string to a TDateTime value.

function *StampToTimeStr(x)* : string
Converts a TDateTime to a string value.

Simple and structured statements, loops:

```
IF expression1 THEN  
  { statement1 }  
[ELSE  
  { statement2 }]
```

If *expression1* is True, then *statement1* is executed; otherwise *statement2* is executed.

IFF(a, s, t)

This is an if statement function. If $a \geq 1$, then *s* as string value is result; otherwise *t* as string value is result.

```
WHILE expression DO  
  { statement }
```

where *expression* returns a Boolean value and *statement* can be a compound statement. The while *statement* executes its constituent statement repeatedly, testing expression before each iteration. As long as *expression* returns True, execution continues.

```
PROCEDURE <ProcedureName>  
  { statement }
```

Create, define procedure.
Attention to stand this definition after **END**.

```
EXEC <ProcedureName>
```

Execute procedure (before **END**.)

```
BREAK
```

Causes the flow of control to exit a while statement.

```
CONTINUE
```

Allows the flow of control to proceed to the next iteration of while statements.

```
EXIT
```

Exits from the current procedure.

```
END.
```

You have to close the procedure with this command. (main block)

Comment: all variables is global.

Example:

```
/*
  MACRO EXAMPLE
  2004.
*/

x:=10; y:=100; z:=1000;
WHILE x<110 DO
{
  IF y<60 THEN { y:=x }
  ELSE
  {
    z:=y-x;
    IF x<=70 THEN {i:=x} ELSE {j:=x}
  }
  IF z>950 THEN { BREAK}
  EXEC Next;
}
END.    /* program end */

PROCEDURE Next
{
  x:=x+1;
  EXIT;
  x:=x+10;
}
```

Usage of DLL functions and parameters in program

Your DLL function structure must be the follow to call in **K Database Magic**:

function DLL_function_name(Arg: variant; var Res : pchar) : integer

The „Arg” parameter is a dynamic array, what the DLL function input parameter. The Res is a character pointer, what it result a value to **K Database Magic**. The return value is not equal 0; another case returns zero therefore the value won't be in Res variable.

1.) If you set DLL function name and source field name (in **database import, export project**), then:
the *Arg[0]* parameter's value is the source field value, type is what you choose (logical, integer, text, etc.)

2.) another case:

the *Arg[0]* (character)

- if the **source** dataset is text file (*.csv), the dataset directory including the colon or backslash that separates the path information from the name and extension
- if the **source** dataset is BDE, the dataset directory or alias name

the *Arg[1]* (character)

The structure of character set is:

[Fileds] (fix charactrs like in .ini file)

<1. field name>=<value>

<2. field name>=<value>

<3. field name>=<value>

...

<n. field name>=<value>

the *Arg[2]* (character)

- if the **destination** dataset is text file (*.csv), the dataset directory including the colon or backslash that separates the path information from the name and extension
- if the **destination** dataset is BDE, the dataset directory or alias name

Example a dll function, write in Delphi:

```
library ConvBool;

uses
  Windows,
  SysUtils,
  Classes;

{$R *.RES}

function L_FBOOL(Arg: variant; var Res : pchar) : integer;
begin
  if UpperCase(Arg[0]) = 'T' then
    Res := 'I'
  else
    Res := 'N';
  result := 1;
end;

function A_DBELLENOR(Arg: variant; var Res : pchar) : integer;
var FTable: TTable;
    vTable: TTable;
    i      : integer;
begin
  result := 1;
  res := '';

  FTable:= TTable.Create(nil);
  vTable:= TTable.Create(nil);
  try
    FTable.DatabaseName := Arg[0];
    FTable.TableName := 'dolg.dbf';
    FTable.Active := True;
    vTable.DatabaseName := Arg[2];
    vTable.TableName := 'BEFLOG';
    vTable.Active := True;

    while not FTable.Eof do
    begin
      with vTable do
      begin
        if Locate('RAKTAR;DATUM', VarArrayOf([FTable.FieldByName('RAKTAR').Text,
          FTable.FieldByName('DATUM').Text]), [loCaseInsensitive]) then Edit
        else Insert;
          try
            for i := 0 to 28 do
              FieldByName('ShBef_ELL').Text := FTable.FieldByName('ShBef_ELL').Text;
              vTable.FieldByName('HIBATLAN').Text := 'T';
            Post;
          except
            Cancel;
            raise;
          end;
        end;
      end;
      vTable.Active := false;
      FTable.Active := False;
    finally
      FTable.Free;
      vTable.Free;
    end;
  end;

exports
  L_FBOOL,
  A_DBELLENOR;

begin
end.
```