

Hdb2Win

Application Cook Book

Version 1.1 – May 2018

1 General remarks

1.1 What contains this document?

This document is a brief description how to create applications for the database management system HDB2Win. HDB2Win includes per default two different applications – PaleoTax and PalCol. An application consists in a set of tables (that are in a certain way connected to each other) and a set of programs to display, analyse and output data. Even if it looks like that, but the applications are not part of the database system, and there are not part of the program. It is comparably easy to create applications on their own, at least the part of the data structure. To analyse and output data is still a more complex matter.

1.2. In what consists an application?

An application consists primarily in a set of interconnected database tables. The tables and their interconnections should follow the model of normalisation. Secondly, the application consists of

- (1) a small describing parameter file,
- (2) a start-up file, and
- (3) a WMF, JPG or BMP image.

Having the tables and the three mentioned files, it is possible to work with the database as an application, to append data, to search for them etc. Only analysing and more complex output operations need some more manual work.

1.3 Which software is needed?

You need a complete installation of Hdb2Win from version 2.4 on (applications such as PaleoTax or PalCol are not needed). A vector graphic program such as CorelDraw helps you to create your own application background.

1.4 Which hardware is needed?

Nothing extraordinary special. Processor speed is not important. A few megabytes of free hard-disk space is needed for the database program, and the space for your data (128 MB of free main memory is enough even for large databases).

2 How to get started?

The creation of a new application consists in four steps:

1. Creating the data structure and creating the tables
2. Modifying the background image (WMF, EMF, JPG, BMP)
3. Writing the application parameter file (APP)
4. Writing the start-up file (PTX).

2.1 Data structure

The design of the data structure is the most crucial part of the application because it describes entirely the problem you want to solve with your database application. Everything else is anyhow only 'technical'. So you need to think thoroughly about the objects you want to record with the program. A database consists of various tables; a table is structured in fields and has records. A field may be the name, address, phone number etc. in an address table, a record corresponds to a person, and another record corresponds to another person.

2.1.1 Design

The database model insists that for all data that belongs to a certain class of objects, separate tables must be created. In a database of addresses, countries, cities, addresses, and persons must be stored in different tables. The person table has beside the information on the person (name, surname, phone number, e-mail) a reference to an item in the address table. The address consists of an optional institution, a street name and house number and a reference to a city table. This table has the postcode and a reference to the city name table. The city name table has the name of the city, the phone area code, and a reference to the table of countries which finally has the country name, the country phone code, and the postal abbreviation. Graphically it looks like this:

Table of Persons - PERSONS	Table of Addresses - ADDRESS	Table of Cities - CITY
Name	Institution*	City postal code*
Surname	Street*	City name ⇒ Table of city names*
Phone	House number	
E-Mail Address	City name ⇒ Table of cities*	
Address ⇒ Table of addresses		

Table of City Names - CITYNAME	Table of Countries - COUNTRY
City name*	Country name*
Country ⇒ Table of countries	Country phone code
Phone	Country postal abbreviation

What is behind this splitting of data? First, the rules say that any single data must not be stored more than once (a single data in our example is the name of a city, an address or a country). Imagine that suddenly the name of a country will change, the country name is stored in various places in the database. In this case the data need to be manually updated which may be a source of errors and inconsistency. The model forbids redundancies, so nothing must be stored more often than only one time. For this reason, any objects which belong to the same type of data are stored in one table.

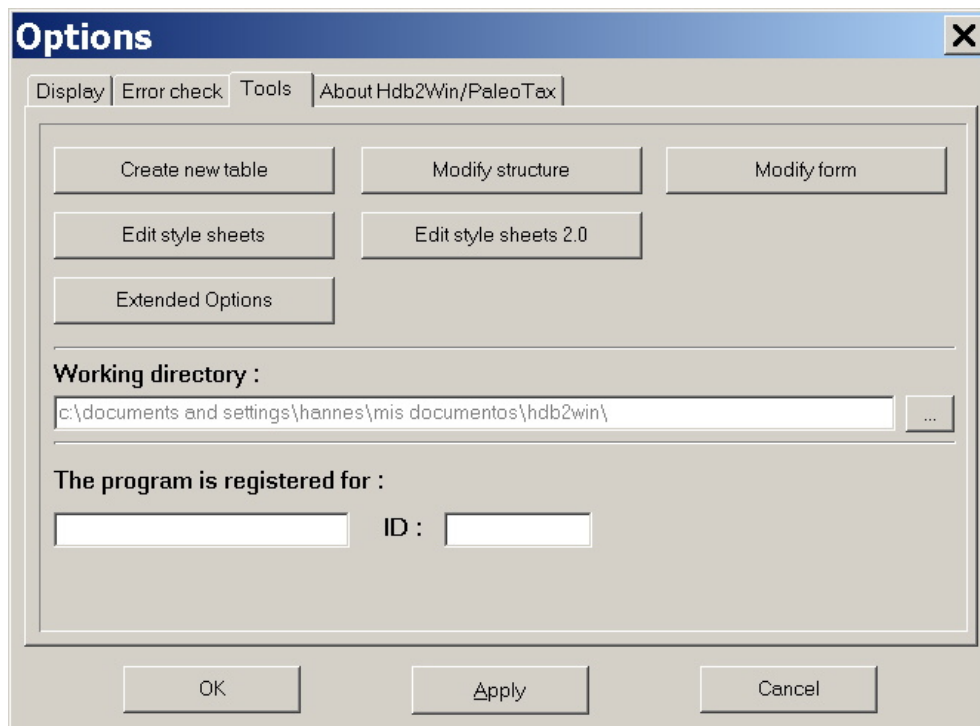
When here is referred to "references" to a table (see the arrows in the above table) it means that one field in a table refers ("is connected to") a specified record in another table. So the address in the main table of persons in the above example is not contained in the table of persons, but in the table of addresses and in the table of the persons is only a link to a record in the table of addresses.

Whereas a database of addresses is somehow trivial, a database for instance of flight booking is more sophisticated. There are travellers, there are flights (from and to airports), there are travels, aircraft, seats in the aircraft, booking classes, prices, special menus etc. etc. The database of addresses consists only of five tables, but a database can easily encompass fifty or hundred tables.

The first step to design a data structures is to collect the different data types (which will later form tables). It is useful to make a list of different data types and to examine how they depend from each other and how they are interconnected, and to list for each data type (table) the data fields. Data fields can be strings (up to 250 characters wide), numbers (integer, without decimals; or real numbers with decimals), logical (allowing only false or true), larger texts, images and references to records of other tables. All other types (links to external files, binary objects) are not part of the DBF standard and are realized in a different way.

2.1.2 Creating tables

After having finished the design, the tables must be created using the database system. Please ensure that your current working directory - preferably C:\Users\\Documents\Hdb2Win\ (Windows 7 onwards) or c:\Documents and Settings\\Documents\Hdb2Win\ (Windows XP) - is correctly assigned (PaleoTax / Options / Tools):



Create a new data folder in your working directory, for instance in our example Persons. The second step is to launch Hdb2Win again, choose PaleoTax, Options, Tools, Create new table.

Choose in the form the above created data path:

Create new table

In: c:\documents and settings\hannes\mis documentos\hdb2win\persons

Filename of the new table (8 chars.)

Alias name (explanation) - German

Alias name (explanation) - English

Alias name (explanation) - Spanish

Start with the tables that do no refer to any other tables (in the above example only the table of the countries). Enter the name of the table: see the list above and apply the names in blue capitals (you do not need to put the .DBF). Give a longer explanation in your preferred language:

Create new table

In: c:\documents and settings\hannes\mis documentos\hdb2win\persons

Filename of the new table (8 chars.)

Alias name (explanation) - German

Alias name (explanation) - English

Alias name (explanation) - Spanish

Click on Execute to start with the creation of the table:

Modify structure

Select a table: (ADDRESS.DBF)

Name of the data field

KEYFLD [//] N = 10

Name:

Alias (Ger.)

Alias (Engl.)

Alias (Span.)

Type of the data field

Character, length:

Number, length, decimals:

Logic Text Graphic

Reference to another table:

Re-Index Key field

Leave the "KEYFLD" unchanged. Give a name to each field (not more than eleven characters for the field name) and give explanations in your language. Define type and length of each field. Interconnected fields require (click on the button with the three points) the selection of another table, the fields which should be shown for selection and the fields which should be shown in the higher ranking table (in the above list these fields are marked with *). Click in "Add field" when ready.

Modify structure [X]

Select a table: (ADDRESS.DBF)

Name of the data field

Name:

Alias (Ger.):

Alias (Engl.):

Alias (Span.):

Type of the data field

Character, length:

Number, length, decimals:

Logic Text Graphic

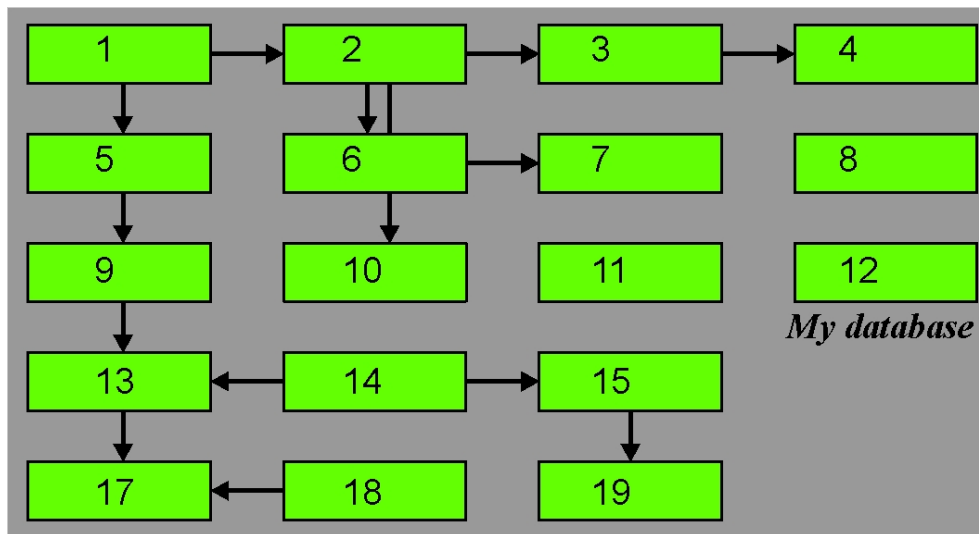
Reference to another table:

Re-Index Key field

Click on "Execute" when ready with the table. You may modify the form, but you can do this later as well. Create the next table. You can always return to a table, add and remove fields.

2.2 Background image

The initial and main form of the application is a image with boxes and arrows. The boxes represent tables that are frequently used, the arrows represent the connections between them. Together with this documentation, files in various formats (cookbook.CDR, *.WMF, *.EMF, *.JPG) are made available that can be modified and saved preferred as WMF or JPG files. See the standard file on the following page.



The numbers are just for your orientation; you will need them later. But in the final version they should be deleted. You may remove boxes, but you should not change their position nor add new boxes. There must remain at least one box. Do not be surprised about the high number of boxes. The address application is a very simple application; there are applications which may have hundred tables of which 15 are really important and often need direct access. So if you have a small application you should plan for every table a box, but if you have an application with many tables, you should only reserve boxes for the most important tables. The boxes are currently numbered from left top to right bottom, so the first row encompasses the boxes one to four, the second five to eight, etc. Please note that box 16 and 20 are not available. You must not put names into the boxes, but you may use the free space beyond the boxes for texts or even images. You may also change the colour of the background and the colour of the boxes. - In the end you should know which table will be in which box because this information is needed for the next step. If a JPG or BMP is created, it should have a resolution of 144dpi what results in a size of 766*414 pixels. JPG is recommended because it is smaller. RGB is recommended.

2.3 Application parameter file (APP)

The application file is a simple ASCII file that briefly describes your application and helps Hdb2Win to identify the tables and to find the background image file. The application file must have an eight letter name with the three letter extension APP.

The file format is simple:

```
keyword,value[ ,value]
```

A keyword word is followed by a comma, a value, and in some cases a second value. There are currently only four key words. Any application file must have all of them!

2.3.1 Main file

The key word 'main file' is followed by the name of the table that has the focus when the application is started; it is usually the table which is most frequently used. For instance:

```
Main file,PERSONS
```

2.3.2 Background image

The key word graph file contains the name of the background image. It may be a WMF or JPG file.

```
Graph file, PERSONS.WMF
```

2.3.3 Id of the application

Any application must have a three letter code which you may define as you want.

```
ID, PER
```

2.3.4 File list

The file list describes which tables are shown in the boxes of the images. So here you have to give the name of the table and its position as described above. The numbers correspond to the numbers in the boxes:

```
File, PERSONS, 5
```

```
File, ADDRESS, 6
```

```
File, CITY, 10
```

```
File, CITYNAME, 14
```

```
File, COUNTRY, 18
```

It is very important that you give here the names of the files you have created. Any mistake will result in an error message.

2.4 Startup file (PTX)

The start-up file is a simple ASCII file which opens the database and connects the database with an application. Start-up file and applications are kept separated because you may have various structurally identical datasets of the same application. In this case you need various start-up files (each database demand its own start-up file) but only one application file. The start-up file must be a eight letter name with a the three letter extension PTX. Actually it is a program file of the Hdb2Win internal interpreter. The start-up file opens the various tables of the database. If one table refers to another table (here, for instance, the table of persons refers to the table of addresses), it is not necessary to open the referred table (addresses) because the table that refers to another table, opens it automatically. But any table to which no other table refers must be opened.

So, all independent tables need be opened. The code word is OPEN:

```
open persons, 7
```

The number 7 is a parameter that should not be modified.

The name of the application should be assigned as well in the start-up file:

```
MOV APP, 'Persons'
```

The structural level must be indicated. This is like a version for the structure and can be zero:

```
MOV SR, 0
```

For the current project, these three lines constitute already the start-up file.

2.5 File location

All three parameter files should be in the data directory together with the database, so normally in C:\Users\\Documents\Hdb2Win\ (Windows 7 onwards) or C:\Documents and Settings\\Documents\Hdb2Win\ (Windows XP).

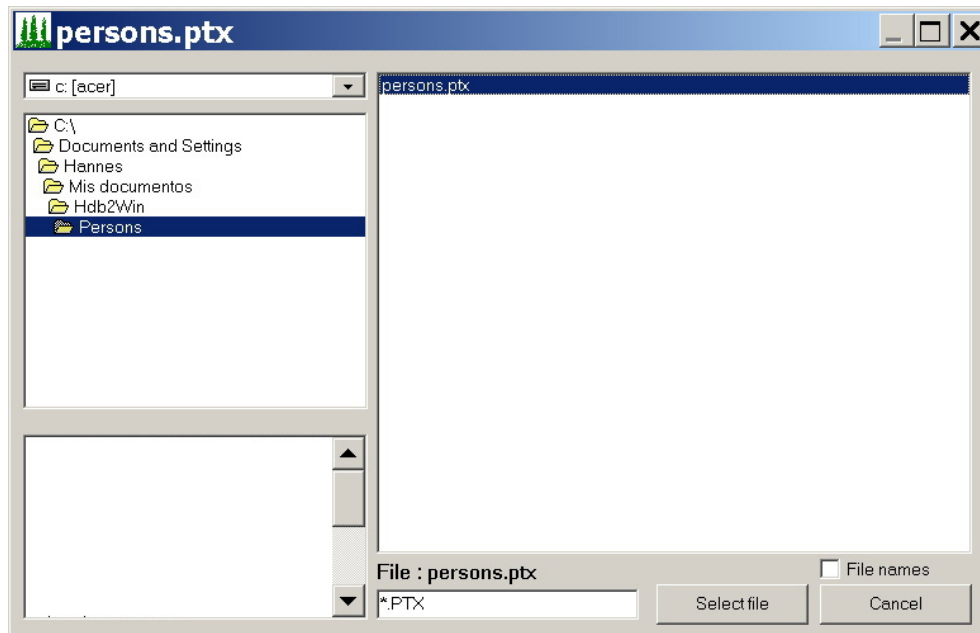
3 Work with the new database

3.1 Register the database

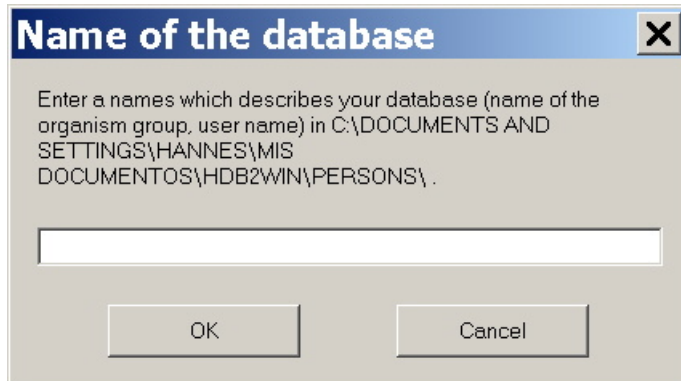
Open Hdb2Win and select PaleoTax. The list of databases will be empty:



To register your new database, you have to click on Search and search in the data directory the startup file persons.ptx:



After you have selected the file, the program will ask you for a real name for the database:

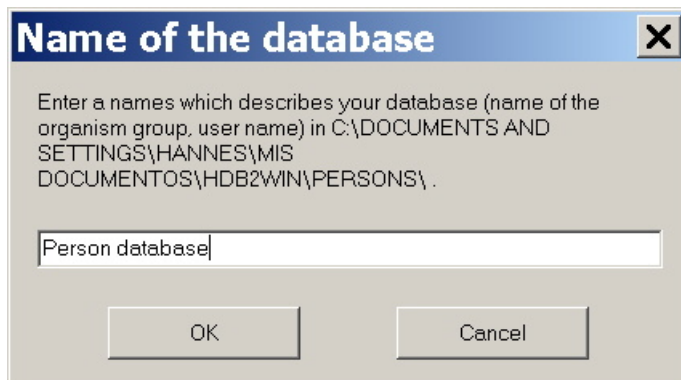


Name of the database [X]

Enter a names which describes your database (name of the organism group, user name) in C:\DOCUMENTS AND SETTINGS\HANNES\MIS DOCUMENTOS\HDB2WIN\PERSONS\ .

OK Cancel

For instance like this:



Name of the database [X]

Enter a names which describes your database (name of the organism group, user name) in C:\DOCUMENTS AND SETTINGS\HANNES\MIS DOCUMENTOS\HDB2WIN\PERSONS\ .

Person database

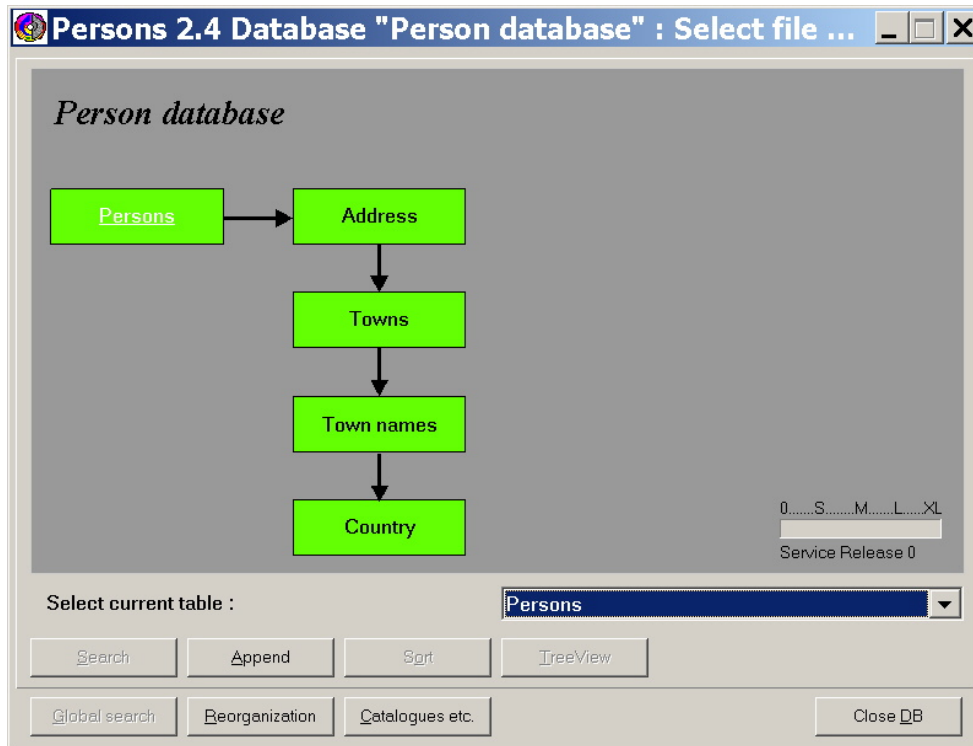
OK Cancel

Now the database appears in the database list. That means that is has been successful registered.



3.2 Open the database

To open the database just click on Open:



4 Limitations

The functionality that offer published applications such as PaleoTax and PalCol are not available in a newly created application. For instance the reorganisation menu is limited; only the creating of the access files and the backup copy is included.

All other functions are hidden in external interpreter programs:

Remove temporary files – DELTEMP.PRF

Various index files – INDEXP.PRF*

Global search – MGI.PRF*

Find unused records – CHCKBASE.PRF*

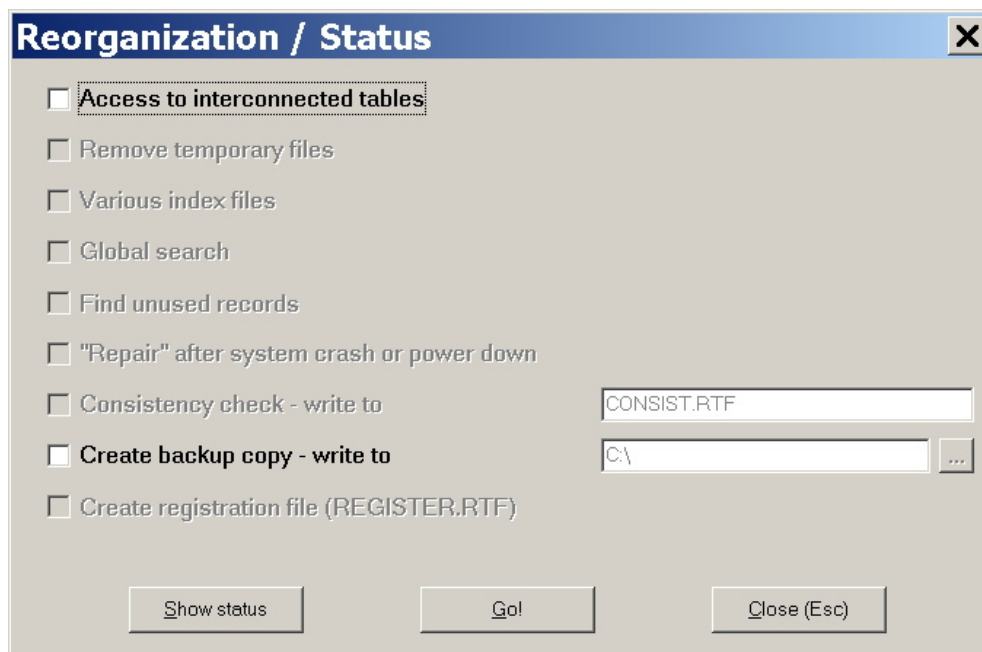
“Repair” ... – TESTINTG.PRF

Consistency check – CONSIST.PRF*

Create registration file – REGISTER.PRF

(*) Some of these programs depend directly on the data structure and need special adaptation.

The same applies to sort and search options.



5 Files

In a separate ZIP archive cb-1.zip you will find a folder Persons that contains the database created as an example. In a second folder Template you will find a template for the background image in four different formats.