

# DB: Open Source Database Program for PalmOS

The db Developer Team

August 14, 2001

*[The current docs are a rudimentary collection of existing files and snippets. I hope to have them shaped up into something readable soon. — mer]*

## 1 Introduction to DB

DB is a flat-file database application for PalmOS-compatible PDAs from companies such as Palm and Handspring. Using DB, you can create databases to store any sort of information that can be stored as records in a database. Several field types are provided for the information in each record.

DB is Free Software provided under the terms of the GNU General Public License (GPL). Read the file "COPYING" that is included in this distribution to learn more about the GNU GPL.

Since DB is under constant development, read the file "NEWS" included in this distribution to learn what major changes have taken place since the previous releases. A more in-depth list of changes is available in the file "ChangeLog" included in this distribution.

**WARNING: While every effort has been made to find and remove bugs from DB, it is still under active development. You use it at your own risk. Since DB is still under active development, there may be bugs which will cause your PDA to crash to the point where a hard reset is required. (Hard resets cause loss of all data on the PDA.) However, many of us find it incredibly useful, and use it on a daily (hourly?) basis.**

### 1.1 Installation

Since DB is available in multiple languages, you must first choose which file will actually be installed to your PDA. Choose the particular file based your desired language:

English     db-en.prc

English     db-en-mobileDB.prc (the MobileDB support compiled inside)

German     db-de.prc

Portuguese db-pt.prc

Italian      `db-it.prc`

French      `db-fr.prc`

Dutch      `db-nl.prc`

You can choose different plugins:

information Plugin/information/information.prc display the number of records, just a sample plugin

JFile3 Plugin/JFile3/JFile3.prc driver for JFile3 support, in alpha

MobileDB Plugin/MobileDB/MobileDB.prc driver for MobileDB support, one of the prc's already contains this support)

DB2 Plugin/DB2/DB2.prc driver for old DB 0.2.x support

The next step is to actually install this file on the PDA. Users of Microsoft Windows and the Macintosh should use the Install Tool that comes with the Palm Desktop software to install one of the above PRC files. Afterwards, initiate a HotSync to complete the installation.

UNIX users should use a program like "pilot-xfer" which is a part of the pilot-link package to install one of the above PRC files. (Other packages like KPilot, JPilot, and gnome-pilot will work just as well.)

## 1.2 Resources (Web and E-Mail)

You can find the latest version of DB at the following Web site:

<http://pilot-db.sourceforge.net/>

You can also join a mailing list for general discussion of DB. To learn how to subscribe to this mailing list, just visit the following Web site:

<http://lists.sourceforge.net/mailman/listinfo/pilot-db-list>

You can post to this list by sending email to the following address:

[pilot-db-list@lists.sourceforge.net](mailto:pilot-db-list@lists.sourceforge.net)

Development of DB has been hosted at SourceForge for some time. SourceForge is a hosting site for many Open Source software projects provided by VA Linux Systems, Inc. DB's particular project page on SourceForge is at the following Web site:

<http://sourceforge.net/projects/pilot-db/>

## 1.3 Building Databases on your Desktop System

This package does not contain the conversion utilities that will take text files and produce DB-format databases. They are available in a separate package called "palm-db-tools". Get version 0.3.2 or higher which supports this version of DB. You can get the latest copy of the palm-db-tools from:

<http://palm-db-tools.sourceforge.net/>

## 2 Usage

### 2.1 Chooser

The database chooser allows you to open and manage all of your databases.

To open a database, tap its name.

To create a new database of any type, tap the New button.

All of the other database operations can be accessed from a database's context menu. Tap the icon next to a database's name to activate the context menu for that database. The operations are:

**Open Read/Edit** Open the database for normal viewing and editing, all data are protected before modification.

**Open Read/Write** Open the database for normal viewing and editing without protection.

**Open Read-Only** Open the database for viewing only. No changes can be made while in read-only mode.

**Info** Activate the information dialog where you can set the category the database should come under in the chooser list.

**Beam** Send this database to another device via the IR port.

**Delete** Remove this database permanently from the device.

**Rename** Change the name of the database.

**Duplicate** Create an identical copy of the database under a different name.

**Create using...** Similar to tapping the New button except the schema of the selected database will be used to initially fill in the design form.

You can customize the database list so that the databases are listed in either alphabetical order or a manual order chosen by you. Use the Display option on the Options menu to choose. In manual mode, press the pen on a database name and drag the database to the desired position in the list.

Use the category selector in the upper-right corner to restrict the database list to only those databases in a particular category. You can set a database's category by choosing Info from the database's context menu.

Advanced option: Choosing the Rescan Databases option from the Advanced menu will force DB to search the device for new databases that it might not know about.

#### 2.1.1 Chooser Create

Enter the title for the new database into the text field.

DB supports more than one database type for various reasons. Use the Type dropdown list to choose what type should be used for the new database.

Note: Not all of the database types support all of DB's features. DB's native type (called DB aptly enough) should be used normally unless you have a compelling reason to use another format.

### **2.1.2 Chooser Display**

Use the Sort Order drop-down list to choose an ordering for the listed databases:

**Alphabetic** The databases are listed in alphabetical order based on the title. Note: Once selected, any manual ordering will be forgotten.

**Manual** There is no particular ordering. You can change the ordering by simply dragging the database to the desired position.

### **2.1.3 Chooser Rename**

Enter the new name for the selected database into the text field.

If a database with that name already exists, you will be notified. Simply choose another name.

### **2.1.4 Chooser Duplicate**

Enter the name for the copy of the selected database into the text field.

DB supports more than one database type for various reasons. Use the Type drop-down list to choose what type should be used for the copy. This is very useful when switching from a legacy type such as the DB 0.2.x type to DB's native type which support all of the newer features.

You can choose to copy the entire source database or just its design. Use the Extent drop-down list to determine what happens.

### **2.1.5 Chooser Info**

Select a category for this database by using the Category drop-down list.

You can hide this database from view by enabling the Private checkbox. The database will be hidden when the device has been set to Hide Private Records via the Security application.

The database type to which this database belongs is displayed on the Type line.

## **2.2 Listview**

The list view displays the records of the database in a simple tabular form.

The ordering and size of the columns can be customized via the Edit List Views... command in the ListBox of the title.

Tap the Find button to display only Records containing the string.

Tap the All button to display all Record.

Tap on the grid's header to sort the records switch the column selected.

The records can be sorted via the Sort command on the Options menu.

The database design such as the number and type of fields can be changed via the Edit Database Design command on the Options menu.

### **2.2.1 Listview Display**

The set of columns displayed can be changed by choosing a different list view from the View drop-down.

### **2.2.2 Listview LFind**

This dialog allows you to search the database for specific strings. Only matching records will be showed in the list view.

Tap the All button on the list view to show all records.

Enable the Case Sensitive Search checkbox if you want case to matter during the search.

Enable the Match Whole Word checkbox if you want the search string to only match against whole words.

Use the Search drop-down to restrict the search to a specific field.

## **2.3 Sort Editor**

Choose which fields to use when sorting the database by adding a row for each field.

When the Sort button is tapped, the first field will be used to determine an order for the records. If there are records with the same value for that field, each additional listed field will be used to determine an order until an order is known or no more fields are listed.

Add or delete a row by tapping the down arrow at the left of a row. A context menu will be displayed which will allow for removing a row and inserting a field before or after the row.

If row has its D column marked, then a sort on that field will be done in descending order (C, B, A) instead of the normal ascending order (A, B, C).

If a row has its CS column marked, then a sort on that field will consider uppercase and lowercase letters to be different. The standard behavior is not to consider case when comparing letters.

## **2.4 Design View**

For new databases, this screen allows the initial structure of the database to be created. For each field in the new database, fill in a row of the display. There is no need to fill in every single row. DB will automatically recognize that the remaining rows are blank and will not use them.

For existing databases, the current database structure will be displayed for review or modification. Some changes to the design of an existing database may be extensive enough to require a complete rebuild of the database. DB will always ask for confirmation before a rebuild when the Done button is tapped.

You can add and delete rows by tapping the down arrow on the left side of a row. This will display a context menu for that row offering choices: Insert before, Insert after, and Remove.

Every field must have a name. This is especially important for new databases since DB will not recognize a row as a field unless you fill in the name.

The popup list on the right-hand side of the display allows you to choose a data type for a field. A data type specifies to DB the type of information that will be stored in the field. There are several types available:

String	plain text
Boolean	true/false, on/off, etc.
Integer	integer numbers between -2147483648 and 2147483647
Date	month, day, and year
Time	hour and minutes
Note	string of 11 characters + note of 500 characters displayed in a memo

## 3 Build Instructions

### 3.1 Requirements

This section describes how to produce the db-LANG.prc binary file from the source code distribution. (LANG will be replaced with a language code.)

The first step is to make sure that you have a recent copy of the prc-tools package installed. The prc-tools package includes a complete compiler toolchain and debugger that support PalmOS. (It is just gcc, binutils, and gdb with PalmOS patches applied.) You will need to install prc-tools version 2.0.92 (2.1pre3) or higher. Follow the instructions at the following Web site to download this version of prc-tools:

<http://prc-tools.sourceforge.net/>

Next, you will need to install a SDK (Software Development Kit) which contains the header files and libraries that are needed to build a PalmOS application. Make sure that you install version 4.0 of the PalmOS SDK. You should also be able to use version 3.5 of SDK. However, in that case, make sure to apply the Update #1 to SDK version 3.5. PalmOS SDK 4.0 can be found at the Web site:

<http://www.palmos.com/dev/tech/tools/sdk40.html>

You will also need a recent copy of PilRC which turns the db.rcp resource file into the binary resources expected by PalmOS. Visit the PilRC home page to download a copy of PilRC. You should download and install version 2.7a or later. The PilRC Web site is at:

<http://www.ardiri.com/index.cfm?redir=palm&cat=pilrc>

To change the configuration feature you need to install the package of autoconf (GNU or Cygwin).

## 3.2 Build

You are now ready to build DB. To build all versions of DB, just run:

```
./configure make
```

You will now see a db-en.prc file which can be installed on your Palm Pilot device using HotSync or pilot-xfer.

A specific language version can be built by running the command:

```
make db-LANG.prc
```

where LANG is the two letter language code for the language that you want. DB currently supports "en" for English, "de" for German, "fr" for French, "nl" for Dutch, and "ja" for Japanese (some problem with this language).

You can clean up all of the files that are created during the build process by running the command:

```
make distclean
```

## 3.3 Configure

Configure can modify the features build in the application. the script "configure" accept different options, to see this options run the command: `./configure --help`

Quickly, the options accepted are:

- `--enable-color` to build a color version.
- `--enable-icons` to build a iconic version.
- `--enable-mobiledb` to add the MobileDB file support.
- `--enable-db2` to add the db 0.2.x file support.
- `--enable-jfile3` to add the JFile3 file support.
- `--disable-sort` to delete the sort editor feature.
- `--disable-plugins` to delete the plugins support.
- `--disable-linkaware` to delete the LinkAware support.

## 3.4 Troubleshooting

If you are some trouble during the compilation began with: `make distclean` and continue with the normal procedure: `./configure make`

# 4 Translating

DB has had support for multiple languages since version 0.21. All of the language-specific strings have been isolated in the separate include files in the "langs" subdirectory to make translation easier.

This file documents how a new language translation can be added.

**Step #1: Create new language file** Look in the langs subdirectory. You will find several files named based on the two letter ISO country code. Find the ISO country language code for your language in the table at the bottom of this file. Replace CODE in all the following sections with the language code from the table.

Copy the file langs/sample.rcp to langs/CODE.rcp. Then add the following line to db.rcp directly after the other include statements at the start of that file:

```
include(CODE.rcp)
```

**Step #2: Translate all the strings** Bring up the file CODE.rcp in your favorite text editor. Find the section bracketed by:

```
TRANSLATION "SAMPLE" BEGIN ... END
```

Change the word SAMPLE to the code of the language that you are providing a translation for. Please use all capital letters for consistency with the other language translations.

Between the BEGIN and END words, you'll find a series of lines that look like this:

```
"Enter string to search for:" = "Enter string to search for:"
```

The first string tells the build system what to look for. The second string is what you need to translate. For example, the German translation turned the above line into this:

```
"Enter string to search for:" = "Suchtext eingeben:"
```

**Step #3: Telling the build system about your translation** Bring up the file Makefile in a text editor. This is the file that controls the compilation of DB. Find the section that looks like this:

```
db-sample.prc: db ts-bin-sample $(BUILDPRC) --output=db-sample.prc
db.def db *.bin
```

Copy these two lines and then change every occurrence of the word sample in the section to your language code. Use lower case for the language code here.

Next, find the section that looks like this:

```
ts-bin-sample: ts-res $(RM_F) *.bin ts-bin* $(PILRC) -q -L SAMPLE
res/res.tmp $(TOUCH) ts-bin-sample
```

Again, copy this section and change every occurrence of sample to the language code. Use lower case except for the one spot where there is a reference to SAMPLE in upper case.

**Step #4: Building a translated copy of DB** At a command prompt, you can build a translated copy of DB using the command:

```
make db-CODE.prc
```

Replace CODE with the code of the language that you want to get a version of DB for.

That's it! :)



## 4.1 ISO Language Codes

aa Afar ab Abkhazian af Afrikaans am Amharic ar Arabic as Assamese ay Aymara az Azerbaijani  
ba Bashkir be Byelorussian bg Bulgarian bh Bihari bi Bislama bn Bengali; Bangla bo Tibetan br Breton  
ca Catalan co Corsican cs Czech cy Welsh  
da danish de german dz Bhutani  
el Greek en English eo Esperanto es Spanish et Estonian eu Basque  
fa Persian fi Finnish fj Fiji fo Faeroese fr French fy Frisian  
ga Irish gd Scots Gaelic gl Galician gn Guarani gu Gujarati  
ha Hausa hi Hindi hr Croatian hu Hungarian hy Armenian  
ia Interlingua ie Interlingue ik Inupiak in Indonesian is Icelandic it Italian iw Hebrew  
ja Japanese ji Yiddish jw Javanese  
ka Georgian kk Kazakh kl Greenlandic km Cambodian kn Kannada ko Korean ks Kashmiri ku Kurdish ky Kirghiz  
la Latin ln Lingala lo Laothian lt Lithuanian lv Latvian, Lettish  
mg Malagasy mi Maori mk Macedonian ml Malayalam mn Mongolian mo Moldavian mr Marathi ms Malay mt Maltese my Burmese  
na Nauru ne Nepali nl Dutch no Norwegian  
oc Occitan om (Afan) Oromo or Oriya  
pa Punjabi pl Polish ps Pashto, Pushto pt Portuguese  
qu Quechua  
rm Rhaeto-Romance rn Kirundi ro Romanian ru Russian rw Kinyarwanda  
sa Sanskrit sd Sindhi sg Sangro sh Serbo-Croatian si Singhalese sk Slovak sl Slovenian sm Samoan sn Shona so Somali sq Albanian sr Serbian ss Siswati st Sesotho su Sudanese sv Swedish sw Swahili  
ta Tamil te Telugu tg Tajik th Thai ti Tigrinya tk Turkmen tl Tagalog tn Setswana to Tonga tr Turkish ts Tsonga tt Tatar tw Twi  
uk Ukrainian ur Urdu uz Uzbek  
vi Vietnamese vo Volapuk  
wo Wolof  
xh Xhosa  
yo Yoruba  
zh Chinese zu Zulu

## 5 DB Database Format

The first database format that DB used was not designed to be extensible. Any additions to the format would break compatibility for previous versions. The new database format for DB is designed to be extensible so that any additional data added to the application information block will not break databases without that data.

### 5.1 Application Information Block

The application information block starts with a small header defining some values that all databases will have. The header is:

```
UInt16 flags UInt16 num_fields
```

The remainder of the data in the app info block is stored as "chunks". Each chunk consists of a simple header followed by the chunk data itself. All of the chunks sit contiguously in the app info block. The header of a chunk is:

```
UInt16 chunk_type UInt16 data_size
```

The next section describes each chunk. For each chunk, the name and chunk\_type value are given followed by a brief description of the chunk and its data.

```
CHUNK_FIELD_NAMES (0)
```

This chunk holds the name of each field in the database. The chunk data consists of a series of zero-terminated strings, one for each name.

```
CHUNK_FIELD_TYPES (1)
```

This chunk holds the type of each field in the database. The chunk data consists of a UInt16 for each field describing the field type:

```
0 - String 1 - Boolean 2 - Integer 3 - Date 4 - Time 5 - Note
```

```
CHUNK_LISTVIEW_DEFINITION (64)
```

This chunk defines a set of columns to be displayed in the list view mode. Each column is defined by the width of the column in the display as well as the field from the database that will be displayed. The chunk data consists of a short header followed by an entry for each column:

```
UInt16 flags UInt16 number_of_columns char name[32]
```

```
UInt16 field_number UInt16 column_width
```

```
CHUNK_LISTVIEW_OPTIONS (65)
```

This chunk stores several of the options that the list view uses. There should be exactly one copy of this chunk in any DB database. Its format is as follows:

```
UInt16 active_view UInt16 first_visible_record
```

```
CHUNK_LFIND_OPTIONS (128)
```

This chunk stores options for the local find dialog. There should be exactly one copy of this chunk in any DB database. Its format is as follows:

```
UInt16 flags
```

### 5.2 Record Structure

Each record in a DB-format database consists of an offset table followed by the actual field data. The offset table exists so that it is extremely easy to find a field in the record.

The offset table consists of one 16-bit unsigned integer (called UInt16 in the PalmOS SDK) for each field. Each entry in the array is the offset of the start of the corresponding field.

The format of each field type is:

FIELD\_TYPE\_STRING (0)

This field type is stored as a null-terminated string.

FIELD\_TYPE\_BOOLEAN (1)

UInt8 value;

This field type stores simple flag values. For true, value will be 1. For false, value will be 0.

FIELD\_TYPE\_INTEGER (2)

Int32 value;

This field type is a 32-bit signed integer.

FIELD\_TYPE\_DATE (3)

UInt16 year; UInt8 month; UInt8 day;

FIELD\_TYPE\_TIME (4)

UInt8 hour; UInt8 minute;

FIELD\_TYPE\_NOTE (5)

Char Title[11]; UInt16 Note\_Offset; Char Note[500];

This field type is a null terminated string followed of the offset of the first char of the Note. The Note is a null terminated string placed at the end of the record. If the note is empty, the offset is null and there are nothing at the end of the record.