

## Portabase

# The Mobile Database

There are many occasions when one would like to have mobile access to data, but converting data between the PDA and desktop systems is often error-fraught. Portabase is a database that runs both on the Sharp Zaurus and on a Linux desktop.

BY FRANK WIEDUWILT

Portabase is a simple database application that does not need a server and is perfectly suited to smaller non-relational databases. Originally programmed for the Sharp Zaurus, the database now also runs on the Linux desktop. We tested the current 1.6 version on a Zaurus with the original Sharp ROM, on a notebook with a current Knoppix version, and with Mandrake 9.0 on a desktop computer.

## Installation

Installing the database on the Zaurus is simple: just download the *portabase\_1.6.arm.ipk* package from the Portabase homepage [1], or copy it from the

subscription CD, and use the Zaurus software installation program, or the Qtopia desktop to install the package. After installing, the icon for launching the program should be displayed in the Qtopia application tab.

Debian users will be happy to hear that a ready-made package is available both from the program homepage and the subscription CD. You can use either *apt-get* or *dpkg* to install the *portabase\_1.6-1.i386.deb* package. You may need to install the *beecrypt* library, which is available from [2].

To launch the program, simply type *portabase&* in a terminal window. Figure 1 shows a program error that occurred on Knoppix 3.2: the menu titles are invisible, although the menus are still operable and the menu items are visible.

Users with other distributions will need to compile Portabase from the sources. Box 1 provides a How-to based on Mandrake 9.0.

And if you need to pass on a Portabase database to a friend or colleague that does not have Linux, there is still no need to worry. The Portabase homepage provides an installation program that will install Portabase on Windows.



## Creating a Database

The desktop and PDA versions look almost identical. The screenshots show the PDA version. After launching the program you are prompted to open or create a new database. To create a new database click on *New* in the menu bar and type a name for the database. This will open the *Columns Editor* where you can define the data fields for your database (Figure 2).

To create a field, select *Add* in the button bar at the bottom of the screen. Type a name for the field in the *Name* box, and select a field type from the *Type* drop-down. Table 1 provides an overview of the field types available in Portabase.

*Default* allows you to assign a default value for a field. *Enum* is a special field type that allows you to specify a list of

## Box 1: Portabase ex Debian

Installing Portabase from the sources is a little involved as the author does not provide a *configure* script, and *Metakit* [3], a library required by the program, needs to be PATCHED. In addition to Portabase and *Metakit*, you will also need the *Beecrypt* library.

Start off by unzipping the *metakit-2.4.8-38.tar.gz* archive using the *tar -xvzf metakit-2.4.8-38.tar.gz* command, and then unzip the Portabase source code archive *portabase-1.6.tar.gz* in your home directory. Change to your home directory and enter *patch -po --input=portabase/metakit-2.4.8.patch* to patch the *Metakit* library, then *cd metakit-2.4.8/builds/*

to change to the *Metakit* directory, and type *../unix/configure* to launch the *configure* script that will create the *MAKEFILES* for the compiler, and compile the library by typing *make*. If you are logged on as *root*, *make install* will now install *Metakit*.

Installing the *beecrypt-2.1.0* is a piece of cake in comparison. Simply unzip the source archive using *tar -xvzf beecrypt-2.1.0.tar.gz*, change to the *beecrypt-2.1.0* directory, and type the following commands *./configure*, *make*, and – again working as *root* – *make install*.

And finally, it is Portabase's turn. You will need to set the *QTDIR* and *QMAKEPEC*

environment variables before compiling; you can enter the *export QTDIR=/usr/lib/qt3/* and *export*

*QMAKEPEC=/usr/lib/qt3/mkspecs/linux-g++/* commands to do so. If you are using a distribution other than Mandrake, you may need to change the path to *Qt*.

After changing to the Portabase source code directory, you can enter */usr/lib/qt3/bin/qmake portabase\_desktop.pro* to create the *Makefiles* and then type *make* to start compiling. Ensure that you are logged in as *root*, and type *make install* to install the program. Typing *portabase&* in a terminal window (Figure 1) will now launch the program.

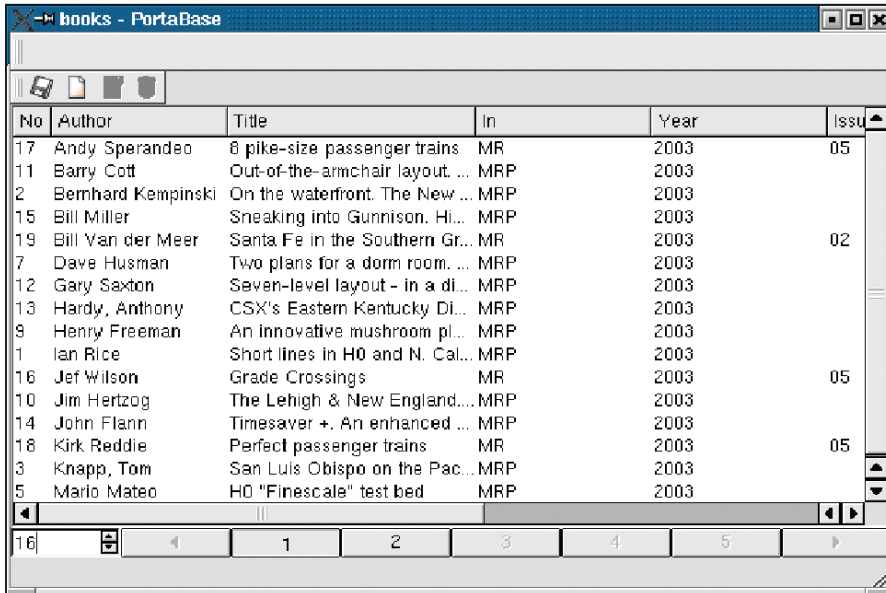


Figure 1: Portabase on KDE 3.1.1

possible entries for the data field. If a list of enumerated types is not available for the current database, Portabase will launch the *Enum Editor* when the Enum type is selected (see Figure 3).

The Enum Editor works just like the *Columns Editor*. After completing the enumerated type, hit *OK* to return to the field definition. You can change the list content later by selecting *File/Edit Enums* in the menu.

Now hit *OK* to save the field definition. While working in the *Columns Editor*, you can use the *Up* and *Down* buttons to change the order of the fields in the data entry mask. Hitting *Edit* will open the field definition for editing. You can then edit the field name and default value, however, the field type cannot be redefined. Selecting *Delete* will remove the

selected field and its content. If you are happy with your database structure, you can select *OK*, to close the *Columns Editor* and return to the record view.

## Data Manipulation

The datasheet view shows the records in the form of a list. A button bar containing a list of numbered buttons is located at the bottom of the screen. Portabase will always display a page of records, but you can use the field at the bottom left of the window to specify the number of records to be displayed on a page.

To create a new record, click on the *New* icon – this is the button with the empty page in the button bar. Doing so will open a window that shows the record in the datasheet view (Figure 4).

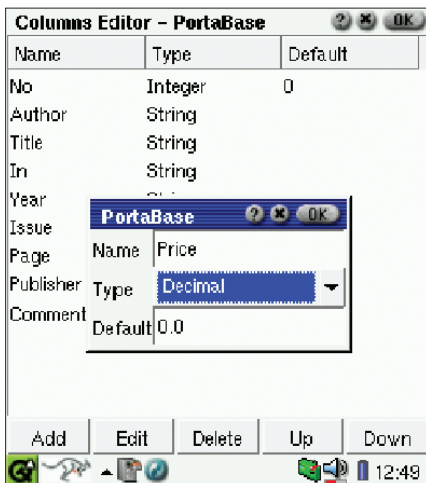


Figure 2: Editing fields

Field Type	Description
String	Text, arbitrary length without newlines
Integer	Integers, range of values from ??? to ???
Decimal	Decimals, range of values from ??? to ???
Boolean	Yes/No values
Note	Longer texts with newlines
Date	Date. Portabase uses the date format specified in your Qtopia preferences.
Time	Time
Enum	Enumerated types. You can define lists with selectable entries for each data base.

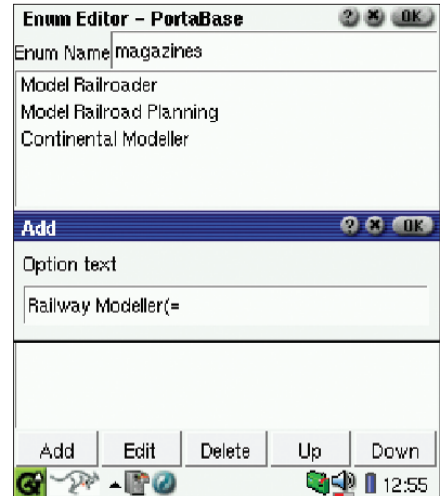


Figure 3: Enumerated types

Fields will be filled with the default values you selected. You can press the [TAB] key to move from field to field. After completing your data entries, close the window by clicking on *OK*. To save the data to a file, click on the *Save* button in the toolbar. Portabase will not automatically save your entries.

## A Question of Views

To avoid cluttered data views the program allows you to select the fields you want to view. To define a *View*, select *View/New* in the menu; this will launch the *View Editor* (Figure 5).

You can select the columns you want to view and use the *Up* and *Down* keys to specify the order of the fields in this view.

## Filtering Records

In contrast to a *View*, which shows specific columns of a datasheet, a filter will

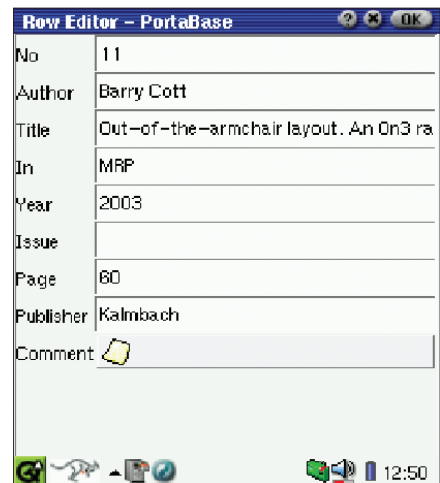


Figure 4: Editing Records

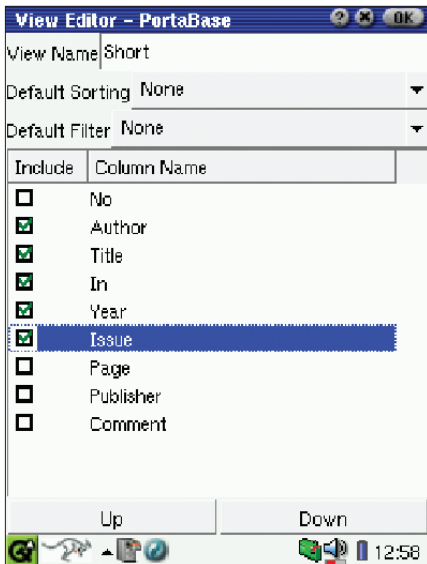


Figure 5: Editing a view

allow you to select the records you want to display. A filter shows only those records that fulfill the conditions.

Portabase has two types of filter. *Quick* filters provide only one condition, whereas normal filters allow multiple conditions. You can create a quick filter by selecting *Filter/Quick* in the menu. This opens a dialog box where you can edit the filter condition (Figure 6).

The upper part of the dialog box contains a list, allowing you to select the field where the filter will be applied. This is *Any Text Column* if you are applying the filter globally. The filter conditions are available in a list on the left of center. Checking *Case Sensitive* will perform a case-specific comparison of the field content with your filter condition.

Finally, use the text box to type your filter condition. Clicking on *OK* will apply the filter and display on those records that match your filter condition.

Filters that need to be repeatedly applied can be saved. To do so, select *Fil-*

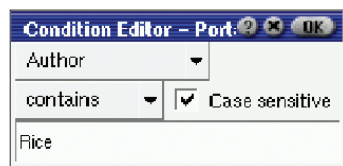


Figure 6: Quick filter

*ter/Add* in the menu, then type a name for the filter and add conditions, as in the *Columns Editor* (Figure 7).

Individual filter conditions are manipulated using the same dialog box as for quick filters.

After clicking on *OK* to save a filter, you will find that it now appears in the filter menu, where you can re-apply it. Filters and views can be combined.

## Keeping Things Tidy

If your records are stored in no discernible order within the database, you might like to tidy things up. To sort the data in a field within the current view, simply select the column header. A single click will sort your records in ascending alphabetical order, clicking once more will select a descending order. You can also sort across multiple fields. To do so, select *Sort/New* in the menu, assign a name to the sorting preferences and specify the fields to sort (Figure 8).

## Data Import and Export

Portabase can read and write data in **CSV** format and in its own XML format. The XML format is discussed at [4]. Data in CSV format need to be **UTF 8** encoded, with each line containing a single record. Strings should be enclosed in double quotes.

Records are always imported into an existing Portabase file that must have the same structure as the import file. To import a file, select *File/Import* in the menu and select the file you want to import in the dialog box.

Importing CSV data tends to work out OK in most cases. However, the program has been known to misinterpret strings as dates and then display the date. Importing a file with 1200 records in CSV format worked fine. The program has no

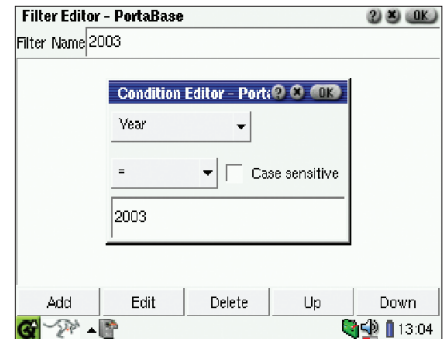


Figure 7: Editing filters

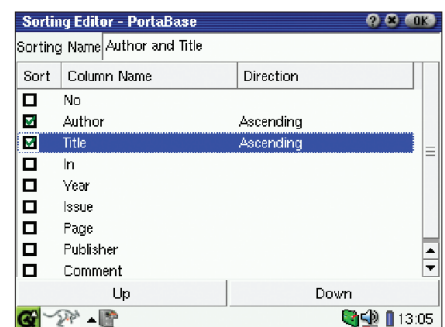


Figure 8: Editing Sorting Preferences

trouble handling databases of this size on the Zaurus, and their were no noticeable lags when working with the file.

You do not need to launch the Portabase GUI to convert a database. You can simply use a terminal window to type a command such as *portabase from-csv books.csv books.pob*, which will convert a CSV file called *books.csv* into the Portabase file, *books.pob*. The *fromxml* converts from XML to Portabase format, and the *toxml* and *tocsv* commands will export a Portabase file to XML or CSV, respectively. To display an overview of the command-line conversion facilities, simply type *portabase -h*.

Portabase created a good impression. With the exception of a display glitch on Knoppix, the program proved to be stable and fast even with larger databases. The online help facility is useful if you find yourself at a loss. ■

## GLOSSARY

**Patch:** A patch contains modifications for a file that the program will insert into the original file.

**Makefile:** A Makefile contains instructions on compiling the source code for the compiler, and on installing the executable program.

**CSV:** Character or Comma Separated Values.

**CSV files** are text-format databases. Each line contains a single record; the data fields are separated by a single character, typically a comma or a tab space.

**UTF:** UTF is a Unicode character set that has a larger selection of characters than standard ANSI, which only allows 256 characters.

## INFO

[1] Portabase: <http://portabase.sourceforge.net>

[2] Beecrypt library: <http://www.virtualunlimited.com/download/>

[3] Metakit library: <http://www.equi4.com/pub/mk/older/metakit-2.4.8-38.tar.gz>

[4] XML format: [http://portabase.sourceforge.net/portabase\\_xml.html](http://portabase.sourceforge.net/portabase_xml.html)