

# THAT'S EASY!

You are writing DB-Applications with Delphi ? Then you should get to know the DBExtender for Delphi !

## DBExtender for Delphi

version 1.1

copyright © 1995,1996 Gfi mbH - Germany

### What is DBExtender ?

DBExtender exist of two new Delphi-components, which will revolutionize the development of databased-applications with Delphi.

All functions, which have to be implemented over and over again when starting a new DB-project, are available right away through these two components.

There are already several new data-aware dialog-object on the market, which replace the original Delphi-components like TTable, TDBEdit, TDBGrid, etc. to achieve a higher performance. But all of those work the same way, the original components do. The components must be linked together. Dialogs contain a multitude of components and hotkeys and speedbuttons do call the methods.

Did you realize, who often the same operations repeat through the single projects ?

This is not necessary within object based environments like Delphi, not if you use DBExtender.

Together with the **standard** Delphi components, DBExtender handles you normal routines automatically. A non-visible component, which relieves you of the hole database specific work.

Nevertheless, DBExtender stays flexible enough, to solve 99% of your specific problems.

To make the functions of the DBExtender "visible", and not only accessible through keyboard, a new navigator, the TDBNavigator comes along. This navigator offers with its 15 new speedbuttons everything, you can expect from a navigator. Through it, you receive direct access to all the dialogs and control-fuctions of the DBExtender.

### How to use the DBExtender ?

You have a "blank" form with all the TTable-, TDataSource-, TDBEdit- and other TDBxxx-components. Place **one** TDBExtender in the form and, if you like, one or more TDBNavigators. Link the navigator(s) to the TDBExtender.

That's it !

When you compile and run your form, all functions on all tables are available right away.

Press [Ctrl]-S, the dialog for searching fieldvalues appears.

Press F7, a table view of the actual table is opened

Press...

The DBExtender recognizes, which table is linked to the focused dialog object and uses that table for the wanted action.

### What are the possibilities of adaptation?

The DBExtender, together with the TDBNavigator, is in charge of 68 properties, 30 methods and 30 events. Not much you might say. But these properties, methods and events are only used for tuning.

The most of the methods are high-power-routines like showing a dialog for filter definition, check and evaluate the user-inputs and activate the BDE-filter with just one method call.

To be always up to date, DBExtender contains a sensitive On-Line-help, which you can call directly from the object inspector or the code editor.

### Overview of the DBExtender-functions.

Following, a short intro to the main DBExtender-functions. And keep in mind: all you need to do to get access to these functions (if not mentioned different) is to place one TDBExtender in your form.

Nothing else !

### Keyboard control within the form

Through the DBExtender, your form get a burst to comfort. 28 hotkeys are made available to every dialog object. This way, the user get access to the dialogs like filtering, searching, etc. and also to some " trivial" things like changing fields via [ENTER] (supplement to TAB), changing records via PgUp, PgDn and much more.

You decide which hotkeys are available to the user by changing a property (design- and runtime).

## Search functions

The DBExtender searches on every key- and non-key field on every table within the form. Substring searches are supported, even on memo-fields!

He offers the possibility to start a new, or continue a previous search. You decide which direction should be used: from the first to the last record or vice versa. (How did say, you have to search from first to the last record ?)

You can start a search, using the predefined dialog. Just click on the nav-button or press [Ctrl]-S and there it is. The dialog uses the table and field as default, which are linked to the current dialog object. A special form of search is the incremental search. Activate the searchmode ([Ctrl]-I or nav-button) and you stay within the form. Every keystroke leads to an expansion of the actual searchstring and a new search. All other functions of the form are still available.

## Bookmark-management

Delphi offers access to the bookmarks via TTables.

We picked up this fine possibility and created a complete management-system for the bookmarks. Use [Ctrl]-K-0 (-9) to set a bookmark and [Ctrl]-Q-0 (-9) to move to the marked record.

Because it can get a little bit difficult to keep in mind which bookmark references which record, the DBExtender contains a bookmark-dialog. This dialog presents the defined bookmarks including the fieldvalues of the marked records. You can then easily find the wanted bookmark and jump to that record. You can also change the order of the bookmarks or delete one. Together with the management-objects, you receive a powerful instrument, allowing your user to define subsets which can be used for sequence-functions.

## LookUp

Of course, Delphi is in charge of components to look up a fieldvalue within another table.

But if you are using Paradox-tables, you have most certainly asked yourself, why do I have to define the lookup-definition (you already defined within the table-structure) over and over again in every form in every lookup-component.

Don't bother about it anymore !

DBExtender knows about the lookup-definitions and automatically offers access to the correct Lookup-form from every component, which is linked to that field ! You need not even place a TTable-, or a TDataSource-component in the Form.

Naturally, the option: 'All corresponding fields' is supported.

To get complete flexibility, an event was created through which you can define lookups during runtime (independent of the table-type).

## Tableview

Do you use a tabbed notebook within a form to offer the user both: one-record-view and tableview ? Save your effort !

The DBExtender offers a tableview for every table within the form. Just press F7 and the table which is linked to the current dialog object will be shown within a new form.

The DBExtender distinguishes between SDI- and MDI-applications. Within SDI-Applications, the tableview appears as a dialog, but if the calling form is a MDI-child, the tableview will be a MDI-child also. Through this you can have an open tableview for every table within your form.

## Filter definition (BDE-Filter)

The DBExtender offers the possibility, to define a filter on every table within the form. He uses the High-Speed-BDE-Filter, which is not accessible through standard-Delphi-components.

And once more: Lean back, everything is already done.

A predefined, easy to use dialog is accessible by keypress. You even can save and load the defined filters.

## Status information and DBException handling

To keep the user up to date, he needs an information source, which tells him the actual tablename, recordnumber, if he is in editmode or insertmode etc. Furthermore he should be informed if an error occurred within the BDE.

The standard procedure to handle the exception would be to use the Application.OnException event, showing a dialogbox. To be honest, this means work, missing flexibility and non-pleasing appearance. The DBExtender offers a new way. Connect a panel for status information and one for messages to the DBExtender (you can also use one panel for both).

Within the StatusPanel, there will be shown the actual tablename, recordnumber, table status etc. The MessagePanel will be used for showing informations on possible actions ('lookup available') or occurred errors ('input required', 'key violation', etc.). Naturally, the DBExtender only interprets

database-exceptions, you will certainly take care of other exceptions yourself. But even in this case, the DBExtender offers help. He defines a FormOnException-event. This way, you can write specific exception handlers (if you want to) for every form of your project.

### **Formglobal edit-mode**

Do you know the problem, you want the best from both worlds ?

Through the AutoEdit-property of the TDataSource it is possible to give the user the freedom, to change every field within every record at any time.

On the other side, you want to protect the user from inserting, changing, or worst deleting a record by accident. Then, you must set the AutoEdit-property to false. But in this case, the user has to enter edit-mode for every record he want to change.

Paradox has a pleasant solution for this problem: the Edit-mode which can be enabled and disabled via F9. If the Edit-mode is active, the user can change, insert and delete as many records as he likes. If the Edit-mode is turned off, all fields react as if there ReadOnly-property is set to True.

The DBExtender takes up this idea and offers you the same solution for your forms !

### **The TDBNavigator**



Lets take a short look at the above mentioned navigator.

Through this you receive access to the TDBExtender-functions via mouse. Do not be frightened. Every single of the 26 buttons can be showed or hidden. Naturally you can switch the navigator to vertical orientation; just set the HorizOrientation-property to false.

Besides its extraordinary appearance, the TDBExtender offers some great inner values. In contrast to its brother, the TDBNavigator, it is linked to an TDBExtender and not to an TDataSource. The big advantage: you do not need one navigator for each table anymore. Use one navigator for all tables. Through its link to the TDBExtender it knows, which table to address for a function initiated by a speedbutton-click.

This even works within MDI-Applications!

Just place one navigator within the frame-form of your application. This one will be automatically linked to the actual table of the active MDI-child. This way, your application looks really professional.

### **Save time and money; test the DBExtender for Delphi now !**

Delphi and Paradox are registered trademarks of Borland International.

Gfi mbH  
Ludwigstr. 15  
65428 Rüsselsheim - Germany  
Phone: +49 6142 965454  
Fax: +49 6142 965455  
eMail: 100415,2071@compuserve.com

The actual trial-version can be found with the Delphi-lib on Compuserve and on World-Wide-Web  
[http://ourworld.compuserve.com/homepages/Gfi\\_mbH](http://ourworld.compuserve.com/homepages/Gfi_mbH)