Contens (V0.99)

What is the DBExtender for Delphi good for?
The DBExtender was created to compensate the shortage on the datasensitive dialogobjects of Delphi and to cover the DB-tasks you have to do over and over again.
To reach that goal, we went an other way. There are a lot of new datasensitive dialogobjects. Several of them are realy good, but all of them have the same problem: they offer a lot of new functions but no real relief, because all functions mußt be called by the programm or they must be linked together.

Actually, the components of Delphi are not to bad. You just have to help them a little bit. That's where the TDBExtender comes into place. He links himself into the messagequeue of the application to reach an extended standard behaviour for the form and its components. Hotkeys are interpreted before they are send to the components and are utilized to activate the extended functions (For example changing a record via [PgUp], [PgDwn], start locating fieldvalues, activating filters etc.). The DBextender knows about **all** tables within the form. Therefore, You do not have to tell the DBExtender which table will be the target for a special action. You just need **one** TDBExtender within a form for all extended functions on all tables and for accessing the ready-build dialogs for locating fieldvalues, filters etc.

To get access to these functions via mouse, a new navigator is part of the DBExtender for Delphi. It possess **14** new buttons (in comparison to the standard-navigator of Delphi). Furthermore, it offers some functions to make it more usable like vertical orientation, spacing between functiongroups, setting the size of the navigator by setting the size of the buttons and much more.
The main advantage of the TDBENavigator is, that you do not need one for each table within a form, you just need **one**. Through the link to the TDBExtender, the navigator knows which table to control.

And within MDI-Applications ? - That's the special clou! You do not need one or more navigators in each form. Just place one navigator within the Frameform of the MDI-Application and the navigator will automatically be switch to the right table of the focused MDI-Child. That's the way to give your applications a professional look.

If you have to write DB-Applications, do not bother arround with "standard-problems". Place one TDBExtender and TDBENavigator within the form and concentrate on the real problems of your application. Programming DB-Apps can be so easy, with the DBExtender for Delphi!

Unit

DBExtend
DBEBMDef

Functions:

Online-Help integrateble within the IDE
Changing focused dialogobjects through [Enter] (TAB-replacement in BOTH directions)
Search on non-indexed fields (incl. search using substrings and memos)
Define the directions of search (also temporary for continued search)
Management of Bookmarks
Tablefiltering (using highspeed BDE-Filters)
Saving and loading tablefilters
Tableview (distinguishes between fsNormal, fsMDIChild)
Memoview (even out of a grid)
Manoeuvre by group
Duplicate fields
Duplicate records
LookUp (like in Paradox, according to the tabledefinition)
Turn Form-Global editing on/off
vertical navigator
Automatic management of navigators within MDI-applications

Showing the table-status within a panel (RecNo, RecCount, Tablename, TableState)
DBExceptionhandling, showing the messages within a panel

= Predefined dialogs

Method LoadTblFilterFromFile

Used by

TDBExtender, Component

Declaration

```
procedure LoadTblFilterFromFile(Tbl : TDBDataSet; FileName : String);
```

Description

Method Clear

Used by
TBMManager, Object

Declaration

```
procedure Clear;
```

Description
The method Clear deletes all TBMItems managed by the TBMManager.

Property DataSource

Used by

TDBENavigator, Component

Declaration

```
property DataSource : TDataSource;
```

Description

Runtime and read-only. This property returns the DataSource, the navigator is actually linked to.

Property VisBtnsRecord

Used by

TDBENavigator, Component

Declaration

```
property VisBtnsRecord : TRecordBtnSet;
```

Description

Through this set you decide, which Speedbuttons are shown within the functiongroup Record.

Possible values put in the set are:

Value	Visible Speedbutton
rnbInsert	Insert record
rnbDelete	Delete record
rnbEdit	Insert editmode
rnbPost	confirm recordchanging
rnbCancel	cancel recordchanging
rnbRefresh	Refresh DataSource

Method SetBookmark

Used by

TBMManager, Object

Declaration

```
procedure SetBookMark(BMNo : Integer);
```

Description

The method SetBookMark creates a new TBMItem (bookmark) on the actual record of the via Table referenced TTable. The number of the bookmark equals the parameter BMNo. If a bookmark with that number exists, it will be deleteed.

Property VisBtnsSearch

Used by

TDBENavigator, Component

Declaration

```
property VisBtnsSearch : TSrchBtnSet;
```

Description

Through this set you decide, which Speedbuttons are shown within the functiongroup Search.

Possible values put in the set are:

Value	Visible Speedbutton
snbSearch	Search on field
snbSearchNext	Continue last search
snbSetBookmark	Set a bookmark on active record
snbGotoBookmark	Move to record, linked to bookmark
snbTblFilter	Edit tablefilter
snbActiveFilter	Activate/Deactivate tablefilter
snbTableView	Switch to tableview

TDBExtender component

Properties Methods Events Hotkeys

Unit

DBExtend

Description

Use the component TDBExtender to enrich all TTables, TQueries, TDataSources and datasensitive dialogobjects within a form.

Place **one** TDBExtender in the form. You have no need to define the used datasources, cause TDBExtender automatically determines the used DataSources and manages them internally.
Set the AllowKeys property to decide, which hotkeys for databasecontrolling will be handled through the TDBExtender. If the focused dialogobject is a TDBEdit, TDBMemo, TDBRadioGroup, TDBCheckBox, TDBGrid or TDBImage, all actions will refer to the DataSource, linked to that dialogobject.

You can control the TDBExtender via mouse through the TDBENavigator. Just like the TNavigator is controlled by the linked TDataSource, the states of the TDBENavigtor-buttons are controlled by the linked TDBExtender.

Define the amount of records to be skipped through a group-manoeuvre by setting the MoveBy property. (Default is 5).

The properties TVReadonly and TVAutoEdit are defaults for tableviews. If you want to control the appearance and the behavior of the tableview, use the event OnTableView.

If you have assigned a panel to the StatusPanel property, you can decide which statusinformation to show by setting the ShowStatusOptions property.

Use MessagePanel, if you want actual informations and DBErrors to be shown within a panel (if HandleDBExceptions is true).

Through the MasterSource property you can define a DataSource, on which hotkeys and TDBENavigator-actions will be applied to, if a non DB-dialogobject has the focus.

If ManageGlobalEdit is set to true, you can toggle a editmode for the whole form via GlobalEdit.

TDBENavigator Component

Properties Methods Events Speedbuttons

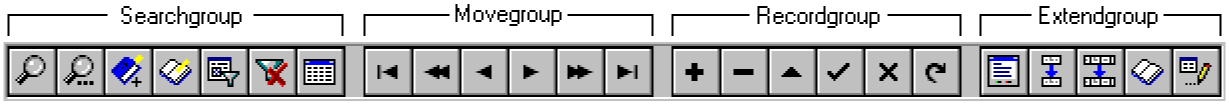
Unit

DBExtend

Description

The TDBENavigator (TDBExtender-Navigator) component is used for controlling the actual datasource, referenced through the linked TDBExtender. Every speedbutton references a hotkey of the TDBExtender. Furthermore, the Navigator offers two more funktions, which can not be accessed through hotkeys: Aktivate/Deactivate filter and Save/Load filter. The latter one can be accessed through a popupmenu. Right-click on the speedbutton for filterediting to open it.

Associate the TDBENavigator via its DBExtender property with a TDBExtender. The single buttons are activated/deactivated through the TDBExtender.



Use the properties VisBtnsSearch, VisBtnsMove, VisBtnsRecord and VisBtnsExtend to decide which speedbuttons will be shown within the functiongroups of the navigator.
Use the properties GroupSpace1, GroupSpace2 and GroupSpace3 to define the distance between the single functiongroups
Use the HorizOrientation property to set the orientation of the navigator to horizontal or vertical.

Method SaveTblFilterToFile

Used by

TDBExtender, Component

Declaration

```
procedure SaveTblFilterToFile(Tbl : TDBDataSet; FileName : String);
```

Description

Methods

AddNavigator
ControlIsDBAware
Create
Destroy
DuplicateFromPrior
DuplicateRecFromPrior
GetBMManager
GetDS_Tbl_Fld
GetDSFromControl
HandleNBBtn
LoadTblFilterFromFile
LocateNextPattern
LocateNextPatternReverse
LocatePattern
LocateViaDlg
LUDlg
MemoView
PanelMessage
RemoveNavigator
SaveTblFilterToFile
SelectBookmark
ShowStatus
TableView
TblFilterEdit

Property DBExtender

Used by

TDBENavigator, Component

Declaration

```
property DBExtender : TDBExtender;
```

Description

Use this property to link the navigator to a TDBExtender. This is similar to the DataSource-linking of a TNavigator.

Method Create

Used by

TDBExtender, Component

Declaration

```
constructor Create(AOwner : TComponent);
```

Description

Method AddNavigator

Used by

TDBExtender, Component

Declaration

```
procedure AddNavigator(Nav: TDBENavigator);
```

Description

Method Destroy

Used by
TDBExtender, Component

Declaration

```
destructor Destroy;
```

Description

Method RemoveNavigator

Used by
TDBExtender, Component

Declaration

```
procedure RemoveNavigator(Nav: TDBENavigator);
```

Description

Method ShowStatus

Used by

TDBExtender, Component

Declaration

```
procedure ShowStatus(DS : TDataSource);
```

Description

Press the Delete-button, to delete the selected bookmark.

Method PanelMessage

Used by

TDBExtender, Component

Declaration

```
procedure PanelMessage( msg : String; AType: TMsgDlgType ); virtual;
```

Description

PanelMessage is used for the output of the message-string on the via MessagePanel defined Panel.
AType defines the type of the message (mtError, mtWarning, mtInformation, etc.).
If MessagePanel is **nil**, PanelMessage will use a MessageDialog (only if AType is mtError or mtWarning)
In case of mtError and mtWarning, there will be a sytembeep together with the messageoutput (onyl is ErrorBeep is TRUE).

Method ControlIsDBAware

Used by

TDBExtender, Component

Declaration

```
function ControlIsDBAware( Control : TControl ) : Boolean;
```

Description

Method HandleNBBtn

Used by

TDBExtender, Component

Declaration

```
procedure HandleNBBtn( Btn : TDBENavigateBtn; Shift: TShiftState; Info
: Longint);
```

Description

Method GetDSFromControl

Used by
TDBExtender, Component

Declaration

```
function GetDSFromControl( AControl : TObject; Var DS : TDataSource) :
Boolean;
```

Description

Method GetDS_Tbl_Fld

Used by
TDBExtender, Component

Declaration

```
function GetDS_Tbl_Fld(    AControl : TObject;
                       Var DS : TDataSource;
                       Var Tbl : TDBDataSet;
                       Var Fld : TField;
                       Var FldNo : Integer;
                       Var IsReadOnly : Boolean):Boolean;
```

Description

Properties

DataSource

DBExtender
VisBtnsSearch
VisBtnsMove
VisBtnsRecord
VisBtnsExtend
GroupSpace1
GroupSpace2
GroupSpace3
HorizOrientation
ButtonWidth
ButtonHeight
Align
DragCursor
DragMode
Enabled
Ctl3D
Hints
ParentCtl3D
ParentShowHint
PopupMenu
ConfirmDelete
ShowHint
TabOrder
TabStop
Visible

Attention : Properties without a jumpmark are equal to those of the TNavigator.

Method DuplicateFromPrior

Used by
TDBExtender, Component

Declaration

```
function DuplicateFromPrior(Fld : TField; Confirm : Boolean) :
Boolean;
```

Description

Method DuplicateRecFromPrior

Used by

TDBExtender, Component

Declaration

```
function DuplicateRecFromPrior(DTbl : TDBDataSet; Confirm :
Boolean):Boolean;
```

Description

DuplicateRecFromPrior generates a **new** record, copies all fields of the actual record and defines it as the new actual record.

Method LocatePattern,

Used by
TDBExtender, Component

Declaration

```
function LocatePattern(    Tbl : TDBDataSet;
                     FldName,
                     SrchStr : String;
                     IgnoreCase,
                     SrchDown : Boolean )  : Boolean;
```

Description

Method LocateNextPattern

Used by

TDBExtender, Component

Declaration

```
function LocateNextPattern( Tbl : TDBDataSet ) : Boolean;
```

Description

Method LocateNextPatternReverse

Used by

TDBExtender, Component

Declaration

```
function LocateNextPatternReverse( Tbl : TDBDataSet ) : Boolean;
```

Description

Method LocateViaDlg

Used by
TDBExtender, Component

Declaration

```
procedure LocateViaDlg( Tbl : TDBDataSet; Fld : TField );
```

Description

Method SelectBookmark

Used by
TDBExtender, Component

Declaration

```
procedure SelectBookmark(Tbl : TDBDataSet);
```

Description

Method LUDlg

Used by
TDBExtender, Component

Declaration

```
procedure LUDlg( DTbl : TDBDataSet;
               LUType : LKUPType;
               TblName : TFileName;
               Var Fld : TField);
```

Description

Method TableView

Used by

TDBExtender, Component

Declaration

```
procedure TableView(Tbl : TDBDataSet; KeepSynchron : Boolean);
```

Description

Properties

OwnerForm

GlobalEdit
AktDataSource
AktFieldHasLU
SrchPossible
NextSrchPossible
TblViewPossible
AktFilterState

ErrorBeep
MoveBy
AllowKeys
ConfirmDelete
ConfDelQuestion
ConfirmOverride
ConfOverQuestion
TVReadonly
TVAutoEdit
HandleDBExceptions
ShowStatusOptions
StatusPanel
MessagePanel
MasterSource
ManageGlobalEdit

Method TblFilterEdit

Used by

TDBExtender, Component

Declaration

```
procedure TblFilterEdit(Tbl : TDBDataSet);
```

Description

Method GetBMManager

Used by
TDBExtender, Component

Declaration
```
function GetBMManager(Tbl : TDBDataSet) : TBMManager;
```

Description
This function returns the bookmarkmanager for a given table.

Events

OnClick;
OnDblClick;
OnDragDrop;
OnDragOver;
OnEndDrag;
OnEnter;
OnExit;
OnResize;

Attention : Events without a jumpmark are equal to those of the TNavigator.

MoveBy, Property

Used by

TDBExtender, Component

Declaration

```
property MoveBy : Integer;
```

Description

This property defines the amount of records to be skipped due to a recordgroupchange. Default is 5.

Events

AppOnMessage
AppOnException
ScreenOnACChange
FormOnActivate
FormOnDeactivate
FormOnKeyPhysical
FormOnFocusControl
FormOnException
OnActiveChanged
OnDataSetChanged
OnDataSetScrolled
OnEditChange
OnEditingChanged
OnTableView

Property AllowKeys

Used by

TDBExtender, Component

Declaration

```
property AllowKeys : TAllowKeys;
```

Description

Through this set you decide, which actions are available to the user. If a values is not included in the set, its hotkey is ignored and the associate speedbutton of the linked TDBENavigators are disabled.

Possible values included in the set:

Value	Meaning
akSelectNext	If true, the user can focus the next dialogobject via [Enter] ([Alt]- or [Ctrl]-[Enter] prior). (There is no equivalent button within the TDBENavigator)
akDBMove	If true, the user can move through the datasource. If false, the actual record can only be changed via programming.
akLookUp	If true, the user can lookup fields, if the field is defined as 'Lookup with help' within the tablestructure.
akSearch	If true, the user can search on fields.
akBookmark	If true, the user can place bookmarks and jump to previously defined bookmarks.
akFilter	If true, the user can edit, activate and deactivate filters.
akDuplicate	If true, the user can duplicate fields or whole records.
akCancel	If true, the user can cancel changes on the actual record.
akDelete	If true, the user can delete a record.
akInsert	If true, the user can insert a new record.
akTableView	If true, the user can switch to a tableview
akGlobalEdit	If true and ManageGlobalEdit is true, the user can toggle the formglobal editmode.

Property OwnerForm

Used by
TDBExtender, Component

Declaration

```
property OwnerForm : TForm;
```

Description

Property GlobalEdit

Used by

TDBExtender, Component

Declaration

```
property GlobalEdit : Boolean;
```

Description

This property toggles the formglobal editmode, if ManageGlobalEdit is true.
For all DataSources, AutoEdit should be TRUE. The global editmode is realized through setting or resetting the Readonly-Properties of the datasensitive dialogobjects within the form. Only those dialogobjects are considered, which have the Readonly-property set to false during designtime.

Property AktDataSource

Used by

TDBExtender, Component

Declaration

```
property AktDataSource: TDataSource;
```

Description

Property AktFieldHasLU

Used by
TDBExtender, Component

Declaration

```
property AktFieldHasLU : Boolean;
```

Description

Property SrchPossible

Used by

TDBExtender, Component

Declaration

```
property SrchPossible : Boolean;
```

Description

Property NextSrchPossible

Used by
TDBExtender, Component

Declaration

```
property NextSrchPossible : Boolean;
```

Description

Property ConfirmDelete

Used by

TDBExtender, Component

Declaration

```
property ConfirmDelete: Boolean;
```

Description

Property ConfDelQuestion

Used by

TDBExtender, Component

Declaration

```
property ConfDelQuestion : String;
```

Description

This property defines the text presented, if the user wants to delete a record. If not set, a standard-text will be used.

Property ConfirmOverride

Used by

TDBExtender, Component

Declaration

```
property ConfirmOverride : Boolean;
```

Description

Property ConfOverQuestion

Used by

TDBExtender, Component

Declaration

```
property ConfOverQuestion : String;
```

Description

This property defines the text presented, if the user wants to duplicate a field but the field already has a value. If not set, a standard-text will be used.

Property TVReadonly

Used by
TDBExtender, Component

Declaration
```
property TVReadonly : Boolean;
```

Description
If a user switches to tableview, this property defines the defaultvalue of the Readonly-property of the TDBGrids within the tableform. The original default is true.

Property TVAutoEdit

Used by

TDBExtender, Component

Declaration

```
property TVAutoEdit : Boolean;
```

Description

If a user switches to tableview, this property defines the defaultvalue of the AutoEdit-property of the TDataSource within the tableform. The original default is false.

Property HandleDBExceptions

Used by

TDBExtender, Component

Declaration

```
property HandleDBExceptions : Boolean;
```

Description

If set to true, the TDBExtender will handle DB-exceptions himself, showing the default BDE-Errorstring within the MessagePanel.

Property ShowStatusOptions

Used by

TDBExtender, Component

Declaration

```
property ShowStatusOptions : TDBShowStatusSet;
```

Description

Property ErrorBeep

Used by
TDBExtender, Component

Declaration
```
property ErrorBeep : Boolean;
```

Description
This property defines, if in case of an DBException handled bei the DBExtender, there will be a system-beep.

Property StatusPanel

Used by

TDBExtender, Component

Declaration

```
property StatusPanel : TPanel;
```

Description

Set this property to an existing TPanel. If done, the statusinformations defined through ShowStatusOptions will be shown there.

Property MessagePanel

Used by

TDBExtender, Component

Declaration

```
property MessagePanel : TPanel;
```

Description

Set this property to an existing TPanel. If done, the actual informations of the datasource will be shown there. If HandleDBExceptions is true, messages about DB-exceptions will also be shown within that panel.

You can use the same panel as for the StatusPanel property.

Property MasterSource

Used by

TDBExtender, Component

Declaration

```
property MasterSource : TDataSource;
```

Description

Through this property, you can define a TDataSource, which will be the target of hotkeys or navbuttonclicks, if the focused dialogobject is not a DB-sensitive object.

Property ManageGlobalEdit

Used by

TDBExtender, Component

Declaration

```
property ManageGlobalEdit : Boolean;
```

Description

This property toggles the controlling mechanism for global editing. If true, global editing can be toggled through the GlobalEdit property. The user switch from and to the editmode via F9 (or with the TDBENavigator), provided that the AllowKeys property contains the akGlobalEdit value.

Type TEditChange

Unit

DBExtend

Declaration

```
TEditChange = procedure(    Sender : TObject;
                            IsEdit : Boolean;
                            var CanChange : Boolean) of object;
```

Description

The parameter IsEdit tell you about the new editmode. Use the CanChange-parameter to forbid the editmodechange.

Event AppOnMessage

Used by

TDBExtender, Component

Declaration

```
property AppOnMessage : TMessageEvent;
```

Description

This property contains the value of the property OnMessage of the object Application, as it was set, when the form was activated (not created!).

Never change the property OnMessage of the Application-object directly, if a TDBExtender is placed in the actual form. You can change the eventfunction by accessing the AppOnMessage-property.

Event AppOnException

Used by

TDBExtender, Component

Declaration

```
property AppOnException : TExceptionEvent;
```

Description

This property contains the value of the property OnException of the object Application, as it was set, when the form was activated (not created!).
Never change the property OnException of the Application-object directly, if a TDBExtender is placed in the actual form. You can change the eventfunction by accessing the AppOnException-property.

Event FormOnActivate

Used by

TDBExtender, Component

Declaration

```
property FormOnActivate: TNotifyEvent;
```

Description

This property contains the value of the property OnActivate of the object Application, as it was set, when the form was created (not activated!).
Never change the property OnActivate of the Application-object directly, if a TDBExtender is placed in a form. You can change the eventfunction by accessing the FormOnActivate-property.

Event FormOnDeactivate

Used by
TDBExtender, Component

Declaration
```
property FormOnDeactivate: TNotifyEvent
```

Description
This property contains the value of the property OnDeactivate of the object Application, as it was set, when the form was created (not activated!).
Never change the property OnDeactivate of the Application-object directly, if a TDBExtender is placed in a form. You can change the eventfunction by accessing the FormOnDeactivate-property.

Event FormOnKeyPhysical

Used by

TDBExtender, Component

Declaration

```
property FormOnKeyPhysical : TKeyPhysical;
```

Description

This property corresponds with the OnKeyDown-event of the form, with three differences :

- this event takes place, even if the KeyPreview-property is false.
- the directionkeys ([Enter], Cursor, etc.) are reported too, even if the focused object does not request them.
- the keypress can be marked as handled (via the var-parameter Handled) and will then be not dispatched to the focused object.

Event FormOnFocusControl

Used by

TDBExtender, Component

Declaration

```
property FormOnFocusControl : TNotifyEvent;
```

Description

The event FormOnFocusControl takes place, if a new control within the form get the focus, before it gets the focus. The parameter Sender contains the object, which will receive the focus.

Event FormOnException

Used by

TDBExtender, Component

Declaration

```
property FormOnException : TExceptionEvent;
```

Description

Through this event you get the opportunity to react directly within a form on exceptions, offering each form its own exceptionhandler.

Event OnActiveChanged

Used by

TDBExtender, Component

Declaration

```
property OnActiveChanged : TDSNotification;
```

Description

Event OnDataSetChanged

Used by
TDBExtender, Component

Declaration

```
property OnDataSetChanged : TDSNotification;
```

Description

Event OnDataSetScrolled

Used by
TDBExtender, Component

Declaration

```
property OnDataSetScrolled : TDSNotification;
```

Description

Event OnEditingChanged

Used by

TDBExtender, Component

Declaration

```
property OnEditingChanged : TDSNotification;
```

Description

Event OnTableView

Used by

TDBExtender, Component

Declaration

```
property OnTableView : TTVNotification;
```

Description

This event takes place, if the user tries to switch into tableview. Use this event, if you want to control the appearance of the tableform by yourself.

Property VisBtnsMove

Used by

TDBENavigator, Component

Declaration

```
property VisBtnsMove : TMoveBtnSet;
```

Description

Through this set you decide, which Speedbuttons are shown within the functiongroup Move.

Possible values put in the set are:

Value	Visible speedbutton
mnbFirst	Move to the first record.
mnbPGroup	Skip a group of records (Direction begin of table)
mnbPrior	Move to the prior record.
mnbNext	Move to the next record.
mnbNGroup	Skip a group of records (Direction end of table)
mnbLast	Move to the last record.

Property VisBtnsExtend

Used by

TDBENavigator, Component

Declaration

```
property VisBtnsExtend : TExtBtnSet;
```

Description

Through this set you decide, which Speedbuttons are shown within the functiongroup Extend.

Possible values put in the set are:

Value	Visible speedbutton
enbDupField	Set the value of the field to that of the prior one
enbDupRecord	Create a new record, filled with the values of the actual one.
enbLookUp	Lookup fieldvalue.
enbGlobalEdit	Toggle global editmode.

Property GroupSpace1

Used by

TDBENavigator, Component

Declaration

```
property GroupSpace1 : Integer;;
```

Description

Use this property to define the distance between the functiongroups Search and Move.

Property GroupSpace2

Used by

TDBENavigator, Component

Declaration

```
property GroupSpace2 : Integer;
```

Description

Use this property to define the distance between the functiongroups Move and Record.

Property GroupSpace3

Used by

TDBENavigator, Component

Declaration

```
property GroupSpace3 : Integer;
```

Description

Use this property to define the distance between the functiongroups Record and Extend.

Property ButtonWidth

Used by

TDBENavigator, Component

Declaration

```
property ButtonWidth : Integer;
```

Description

Use this property to set the width of each speedbutton. The total width of the component will be adapted.

Methods

Create
Destroy
SetBounds
BtnClick

Attention : Methods without a jumpmark are equal to those of the TNavigator.

Method BtnClick

Used by

TDBENavigator, Component

Declaration

```
procedure BtnClick(Index: TDBENavigateBtn);
```

Description

This procedure simulates the click on a speedbutton and therefore calls the referenced action.

Type TAllowKeys

Unit

DBExtend

Declaration

```
TAllowKey  = (  akSelectNext, akDBMove, akLookUp, akSearch,
akBookmark, akFilter,
              akDuplicate, akCancel, akDelete, akInsert, akTableView,
              akGlobalEdit );

TAllowKeys = set of TAllowKey;
```

Description

TAllowKeys is the amount of values, the property AllowKeys can have.

DBExtend Unit

Objects

TDBExtender
TDBENavigator

Types

TAllowKey
TAllowKeys
TDBENavigateBtn
TDBShowStatus
TDBShowStatusSet
TDSNotification
TExtNavBtn
TExtBtnSet
TKeyPhysical
TMoveNavBtn
TMoveBtnSet
TRecordNavBtn
TRecordBtnSet
TSrchNavBtn
TSrchBtnSet
TTVNotification
TFilterState

Exceptions

EDBEFilter

Constants

Functions

GetDuplicateTable

Type TDBShowStatusSet

Unit

DBExtend

Declaration

```
TDBShowStatus = (ssRecNo, ssRecCount, ssTblName, ssEdit, ssInsert);

TDBShowStatusSet = set of TDBShowStatus;
```

Description

Type TDBENavigateBtn

Unit

DBExtend

Declaration

```
TDBENavigateBtn = (   nbSearch, nbSearchNext, nbSetBookmark,
nbGotoBookmark,
                  nbTblFilter, nbActiveFilter, nbTableView,
                  nbFirst, nbPGroup, nbPrior, nbNext, nbNGroup,
nbLast,
                  nbInsert, nbDelete, nbEdit, nbPost, nbCancel,
nbRefresh,
                  nbMemoView, nbDupField, nbDupRecord, nbLookUp,
nbGlobalEdit);
```

Description

Type LKUPType

Unit

DBiTypes

Declaration

```
LKUPType = (           { Paradox Lookup type }
    lkupNONE,          { Has no lookup }
    lkupPRIVATE,       { Just Current Field + Private }
    lkupALLCORRESP,    { All Corresponding + No Help }
    lkupHELP,          { Just Current Fld + Help and Fill }
    lkupALLCORRESPHELP { All Corresponging + Help }
    );
```

Description

Type TKeyPhysical

Unit

DBExtend

Declaration

```
TKeyPhysical = procedure(  Key: Word;
                           Shift: TShiftState;
                           Var Handled : Boolean) of object;
```

Description

Type TDSNotification

Unit

DBExtend

Declaration

```
TDSNotification = procedure( Sender : TObject; DS : TDataSource) of
object;
```

Description

Type TTVNotification

Unit

DBExtend

Declaration

```
TTVNotification = procedure(     Sender : TObject;
                           Var aBorderStyle : TFormBorderStyle;
                           Var aFormStyle : TFormStyle;
                           Var isReadonly : Boolean;
                           Var isAutoEdit : Boolean) of object;
```

Description

Type TSrchBtnSet

Unit
DBExtend

Declaration
```
TSrchNavBtn = (snbSearch, snbSearchNext, snbSetBookmark, 
snbGotoBookmark,
               snbTblFilter, snbActiveFilter, snbTableView);

TSrchBtnSet = Set of TSrchNavBtn;
```

Description
TSrchBtnSet is the amount of values, the property VisBtnsSearch can have.

Type TMoveBtnSet

Unit

DBExtend

Declaration

```
TMoveNavBtn = (mnbFirst, mnbPGroup, mnbPrior, mnbNext, mnbNGroup,
mnbLast);

TMoveBtnSet = set of TMoveNavBtn;
```

Description

Type TRecordBtnSet

Unit

DBExtend

Declaration

```
TRecordNavBtn = (rnbInsert, rnbDelete, rnbEdit, rnbPost, rnbCancel,
rnbRefresh);

TRecordBtnSet = set of TRecordNavBtn;
```

Description

Type TExtBtnSet

Unit

DBExtend

Declaration

```
TExtNavBtn = (enbMemoView, enbDupField, enbDupRecord, enbLookUp,
enbGlobalEdit);

TExtBtnSet = set of TExtNavBtn;
```

Description

Event ScreenOnACChange

Used by

TDBExtender, Component

Declaration

```
property ScreenOnACChange : TNotifyEvent;
```

Description

This property contains the value of the property OnActiveControlChange of the object Screen, as it was set, when the form was activated (not created!).
Never change the property OnActiveControlChange of the Screen-object directly, if a TDBExtender is placed in the actual form. You can change the eventfunction by accessing the ScreenOnACChange-property.

Function GetDuplicateTable

Unit

DBExtend

Declaration

```
function GetDuplicateTable(AOwner : TComponent; STbl : TTable) :
TTable;
```

Description

Hotkeys of the TDBExtender

Through this hotkeys, the user accesses the database-functions of the TDBExtender. A function is accessible, if :

- the focused object is a datasensitive one or
- the property MasterSource is set to a valid TDataSource,
- the property AllowKeys allows that action.

Key	Action
[Enter]	Focus the next dialogobject (equal to [TAB])
[Alt]-[Enter]	Focus the prior dialogobject (equal to [Shift]-[TAB])
[Ctrl]-[Enter]	CR in memofields
[Ctrl]-S	Opens the dialog for searching on fields
F3	Continues the last search
[Ctrl]-F3	Continues the last search with opposite direction
[Ctrl]-K-n	Set a bookmark (n:1-9)
[Ctrl]-K-0	Set a bookmark. The number of the bookmark will be the next one.
[Ctrl]-Q-n	Goto bookmark n (n:1-9)
[Ctrl]-Q-0	Opens a dialog for bookmarkmanagement
[Ctrl]-F	Opens a dialog for editing filters
F7	Switches to tableview
[Ctrl]-F7	Switches to tableview, the table of the TVGrid and the form are synchronized
[Ctrl]-PageUp	Move to first record
[Shift]-PageUp	Skip a group of records (Direction begin of table)
PageUp	Move to prior record
PageDown	Move to next record
[Shift]-PageDown	Skip a group of records (Direction begin of table)
[Ctrl]-PageDown	Move to last record
[Ins]	Insert new record
[Ctrl]-[Del]	Delete record
[Ctrl]-Back	Cancel changes on record
F2	Switches to memoview
[Ctrl]-D	Duplicate field (takeover from prior record)
[Ctrl]-[Shift]-D	Duplicate record (creates new record with values equal to actual record)
[Ctrl]-[Space]	Opens lookup-dialog
F9	Toggle global editmode

Speedbuttons TDBENavigator

These speedbuttons offer the access to the TDBExtender-functions via mouse. If a button is en- or disabled is controlled by the linked TDBExtender.

Button	Action
	Opens the dialog for searching on fields
	Continues the last search
-[Ctrl]	Continues the last search with opposite direction
	Set bookmark on actual record
	Opens a dialog for bookmarkmanagement
	Opens a dialog for editing filters
-Rightclick	Opens a popup-menu for saving and loading filterdefinitions
	Activate/Deactivate filter
	Switches to tableview
-[Ctrl]	Switches to tableview, the table of the TVGrid and the form are synchronized
	Move to first record
	Skip a group of records (Direction begin of table)
	Move to prior record
	Move to next record
	Skip a group of records (Direction begin of table)
	Move to last record
	Insert new record
	Delete record
	Cancel changes on record
	Refresh the DataSource
	Switches to memoview
	Duplicate field (takeover from prior record)
	Duplicate record (creates new record with values equal to actual record)
	Opens lookup-dialog
	Toggle global editmode

Tableview

Within a tableview, all records of the, through the focused dialogobject referenced, DataSet are shown through a TDBGrid within a own form.

There are two kinds of tableviews: the normal and the synchronized. Since the normal tableview uses its own TTable for showing the records, the synchronized uses the same TTable as the dialogobject from which the tableview was initiated. Therefore a change of the actual record within the tableview causes a change in the ownerform too.

Type TFilterState

Unit

DBExtend

Declaration

```
TFilterState = (ftsNA, ftsDropped, ftsAdded, ftsActive);
```

Description

TFilterState declares the values, the property AktFilterState of the TDBExtender component can have.

Property AktFilterState

Used by

TDBExtender, Component

Declaration

```
property AktFilterState : TFilterState;
```

Description

Runtime and readonly. AktFilterState presents the filterstatus of the actual DataSource.

Property TblViewPossible

Used by
TDBExtender, Component

Declaration

```
property TblViewPossible : Boolean;
```

Description

Property HorizOrientation

Used by

TDBENavigator, Component

Declaration

```
property HorizOrientation : Boolean;
```

Description

Use this property to set the orientation of the navigator. Turn it to false and the navigator will be shown in vertical orientation.

Property ButtonHeight

Used by

TDBENavigator, Component

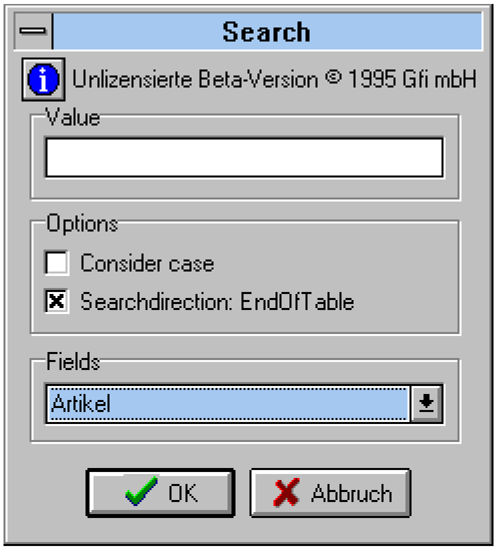
Declaration

```
property ButtonHeight : Integer;
```

Description

Use this property to set the height of each speedbutton. The total height of the component will be adapted.

Searchdialog



Unlizensierte Beta-Version © 1995 Gfi mbH

Value

Options

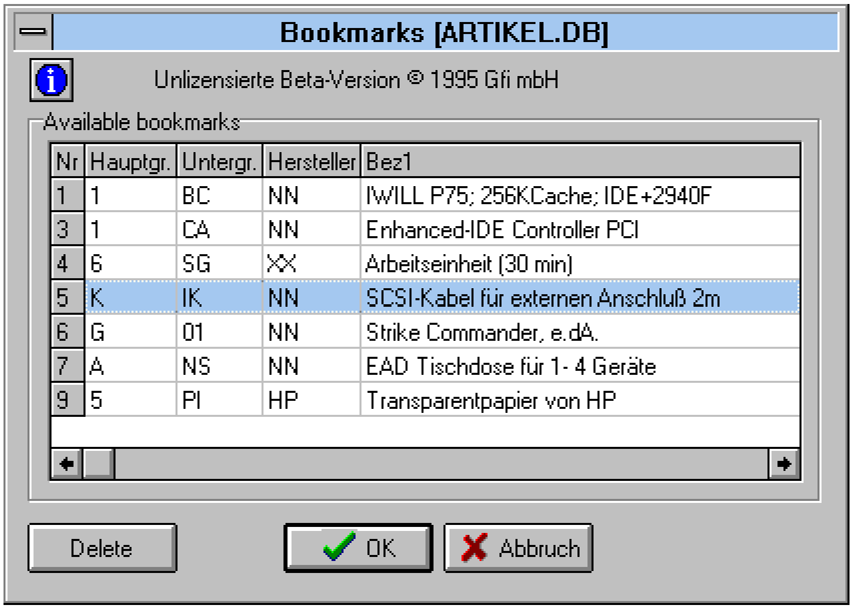
☐ Consider case

☒ Searchdirection: EndOfTable

Fields: Artikel

This dialog appears, if the user clicks on the speedbutton for searching or presses [Ctrl]-S.
Click on the different areas of the bitmap to get more information.

Bookmarkmanagement dialog



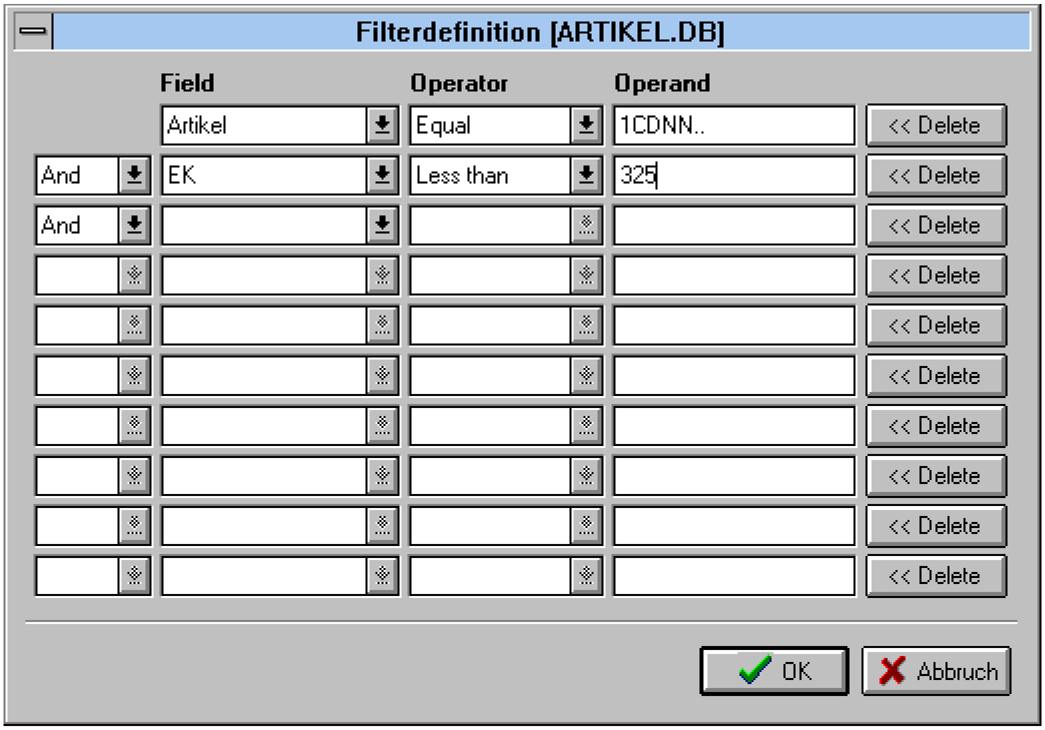
Unlizensierte Beta-Version © 1995 Gfi mbH

Nr	Hauptgr.	Untergr.	Hersteller	Bez1
1	1	BC	NN	IWILL P75; 256KCache; IDE+2940F
3	1	CA	NN	Enhanced-IDE Controller PCI
4	6	SG	XX	Arbeitseinheit (30 min)
5	K	IK	NN	SCSI-Kabel für externen Anschluß 2m
6	G	01	NN	Strike Commander, e.dA.
7	A	NS	NN	EAD Tischdose für 1- 4 Geräte
9	5	PI	HP	Transparentpapier von HP

This dialog appears, if the user clicks on the speedbutton for bookmarkmanagement   or presses [Ctrl]-Q 0.

Click on the different areas of the bitmap to get more information.

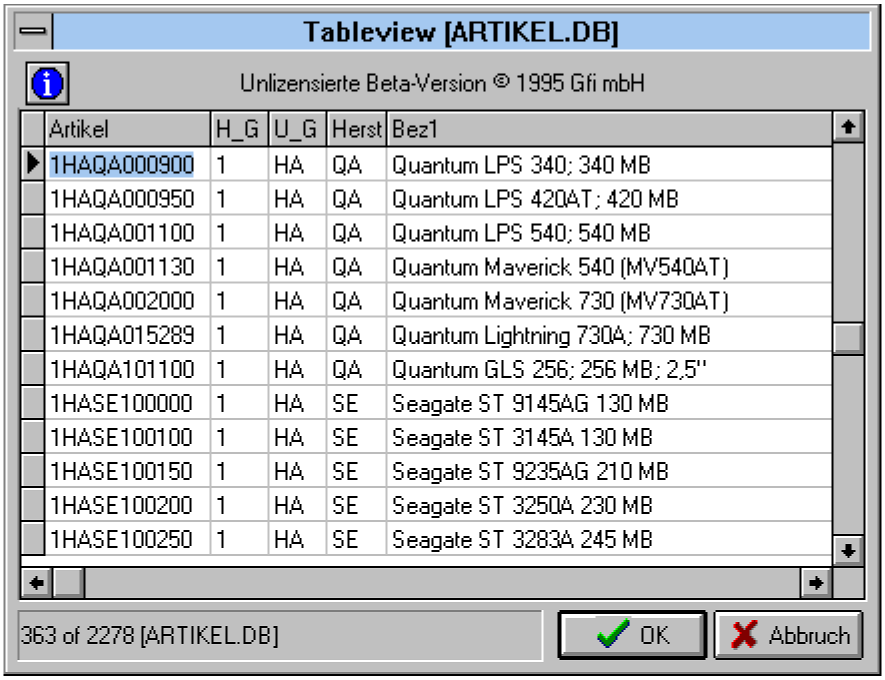
Filterdefinition dialog



	Field	Operator	Operand
	Artikel	Equal	1CDNN..
And	EK	Less than	325
And			

This dialog appears, if the user clicks on the speedbutton for filterediting or presses [Ctrl]-F.
Click on the different areas of the bitmap to get more information.

Tableview form



Tableview [ARTIKEL.DB]

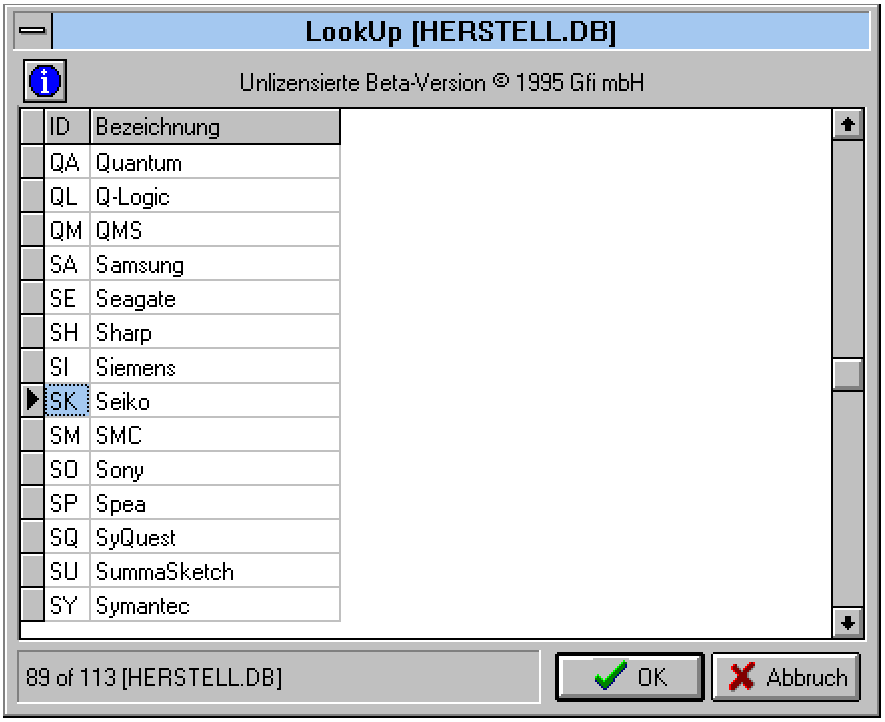
Unlizensierte Beta-Version © 1995 Gfi mbH

Artikel	H_G	U_G	Herst	Bez1
1HAQA000900	1	HA	QA	Quantum LPS 340; 340 MB
1HAQA000950	1	HA	QA	Quantum LPS 420AT; 420 MB
1HAQA001100	1	HA	QA	Quantum LPS 540; 540 MB
1HAQA001130	1	HA	QA	Quantum Maverick 540 (MV540AT)
1HAQA002000	1	HA	QA	Quantum Maverick 730 (MV730AT)
1HAQA015289	1	HA	QA	Quantum Lightning 730A; 730 MB
1HAQA101100	1	HA	QA	Quantum GLS 256; 256 MB; 2,5"
1HASE100000	1	HA	SE	Seagate ST 9145AG 130 MB
1HASE100100	1	HA	SE	Seagate ST 3145A 130 MB
1HASE100150	1	HA	SE	Seagate ST 9235AG 210 MB
1HASE100200	1	HA	SE	Seagate ST 3250A 230 MB
1HASE100250	1	HA	SE	Seagate ST 3283A 245 MB

363 of 2278 [ARTIKEL.DB]

This form appears, if the user clicks on the speedbutton for tableview or presses F7.
The appearance of the form depends on the owner-form. If the FormStyle of the owner-form is fsMDIChild, then the tableview itself will be shown as an MDIChild. Otherwise the tableview will be presented as a dialog.
If the form is an MDIChild, you can have multiple tableviews at the same time and switch between owner-form and tableview, without closing the tableview.

Lookup dialog



LookUp [HERSTELL.DB]

Unlizensierte Beta-Version © 1995 Gfi mbH

ID	Bezeichnung
QA	Quantum
QL	Q-Logic
QM	QMS
SA	Samsung
SE	Seagate
SH	Sharp
SI	Siemens
SK	Seiko
SM	SMC
SO	Sony
SP	Spea
SQ	SyQuest
SU	SummaSketch
SY	Symantec

89 of 113 [HERSTELL.DB]

This form appears, if the focused object has a lookup defined for it (during tabledefinition) and the user clicks on the speedbutton for lookup or presses [Ctrl]-[Space].
If OK is clicked, the fieldvalue of the selected record will be taken over.

Enter the value, you want to search for.
As a wildcard, '..' can be used. This wildcard replaces any amount of any character. A wildcard for a single character is not implemented.
Since all fieldtypes are internally interpreted as strings, you can work with substrings on date- and number-fields too.
Within memos, only the first 64000 characters are used.

Example: The entry '..ext..' would find: 'extract', 'DBExtender', 'text', etc.

Through the option 'consider case' you decide, whether the case of the characters will be important for the search, or not.

Through the option 'Searchdirection: EndOfTable' you decide, whether the search will start at the beginning of the table and searches downwards or opposite. If marked, the search will find the first appearance of the value, otherwise it will find the last.

Through the combobox fields you select the field to search on.

The listbox 'available bookmarks' contains all defined bookmarks for the table.
Each line shows the number of the bookmark and the indexvalues of the corresponding record.

The OK button causes a move to record, referenced through the selected bookmark. If the record can not be found, an exception will be raised.

Through the combobox 'link' you choose the logical connection between two lines of the filter.

Use the combobox 'field' to choose the field, the filterexpression will be assigned to.

Through the combobox 'operator' you choose the operator combining the field and the value. Possible operators are: 'empty', 'not empty', 'equal', 'not equal', 'greater then', 'greater equal', 'less then' and 'less equal'.

Within the editfield 'operant', you insert the filtervalue, which will be combined through the operator with the tablefield.
Using a filter, you are in need to insert a valid value, according to the fieldtype of the referenced field (in contrast to the search on fields).
If you have chosen 'equal' as the operator, you also can use substringfiltering (only on stringtyped-fields).
Use '..' as a wildcard. The wildcard can only be placed at the end of the string, a wildcard in front of the string is not possible.

The button 'delete' clears the line of the filterdefinition.

The OK button sets and activates the defined filter.

TBMManager Object

Properties Methods

Unit

DBEBMDef

Description

The TBMManager-object is used to manage all bookmarks, that are defined on a table. The TDBExtender owns a TBMManager for each table.
The TBMManger is similar to a sorted list. The items in the list are of the class TBMItem (the object, that holds the bookmark). The list is sorted by the bookmarknumber, each TBMItem is referenced by.
Normally, you would not access the TBMItems directly, but you would use the properties and methods of the manager for setting, deleting and going to a bookmark.

Method DeleteBookMark

Used by
TBMManager, Object

Declaration
```
procedure DeleteBookMark(BMNo : Integer);
```

Description
The method DeleteBookMark deletes the TBMItem, referenced by the number BMNo. Nothing happens, if there is no TBMItem with that number.

Property BMItems

Used by

TBMManager, Object

Declaration

```
property BMItems[Index : Integer] : TBMItem;
```

Description

ReadOnly. The property BMItems is the default-property of the TBMManger. Through this property, you get access to all defined BMItems of the manager. The list is sorted by the bookmarknumber of the TBMItems.

Methods

AddBookMark
Create
Clear
DeleteBookMark
Destroy
FindBMbyNo
GotoMark
GetIndexNames
IndexOfBM
MoveBMNo
SetBookMark

Property Table

Used by
TBMManager, Object

Declaration
```
property Table : TDBDataSet;
```

Description
The property Table defines the table, the bookmarks will be managed for. Changing this property leeds to the delete of all bokkmarks defined for the table.

Properties

- Table
- BMItems
- Count

Method AddBookMark

Used by

TBMManager, Object

Declaration

```
function AddBookMark : Integer;
```

Description

The method AddBookmark creates a new bookmark, based on the actual record corresponding to the via Table referenced TTable. The returncode of the function is the number of the created TBMItem. If you would like to create a new bookmark, defining the number of the bookmark yourself, you have to call the SetBookMark method.

Property Count

Used by

TBMManager, Object

Declaration

```
property Count : Integer;
```

Description

Readonly. This property supplies you with the amount of TBMItems, defined within the manager.

$#Method Clear

Used by
TBMManager, Object

Declaration

```
procedure Clear;
```

Description
This method deletes all TBMItems, defined by the manager.

$#Method DeleteBookMark

Used by
TBMManager, Object

Declaration

```
procedure DeleteBookMark(BMNo: Integer);
```

Description
This method deletes the TBMItem, referenced by the bookmarknumber BMNo. If there is no TBMItem with that number, nothing will happen.

Method FindBMbyNo

Used by

TBMManager, Object

Declaration

```
function FindBMbyNo(BMNo:Integer; var BM:TBMItem) : Integer;
```

Description

The method FindBMbyNo returns the TBMItem within the BM-parameter, referenced by the bookmarknumber BMNo. The returncode of the function supplies you with the indexnumber of the TBMItem within the list. If no TBMItem with the BMNo-number can be found, BM equals nil and the returncode is -1.

Method GotoMark

Used by

TBMManager, Object

Declaration

```
function GotoMark(BMNo:Integer): Boolean;
```

Description

The method GotoMark move the table, referenced by the Table property, to the record, referenced by the TBMItem with the bookmarknumber BMNo. The returncode is true, if everything went fine. If there is no TBMItem with a bookmarknumber BMNo or the navigation within the table went wrong, the returncode is false (no exception will be raised!).

Method GetIndexNames

Used by

TBMManager, Object

Declaration

```
procedure GetIndexNames(var IdxNames : TStringList);
```

Description

The method GetIndexNames fills the stringlist IdxNames with the fieldnames of the stored fieldvalues. The fieldnames correspond with the fields, shown within the Lesezeichenbookmark-dialog. The difference between this method and the TTable-method GetIndexNames is, that if no Index is defined for a table, the first MaxCount4NonIdx fields of the table are used.

Constant MaxCount4NonIdx

Used by

TBMManager, Object

Declaration

```
MaxCount4NonIdx = 5;
```

Description

The constant MaxCount4NonIdx defines the amount of fields, used by the Lesezeichendialog, if no Index is defined for the referenced table.

Method IndexOfBM

Used by

TBMManager, Object

Declaration

```
function IndexOfBM(BMNo:Integer): Integer;
```

Description

The method IndexOfBM supplies you with the indexnumber of the TBMItem with the bookmarknumber BMNo. This indexnumber can be used with the BMItems-property. If no TBMItem with that BMNo can be found, -1 will be returned.

Method MoveBMNo

Used by

TBMManager, Object

Declaration

```
procedure MoveBMNo(CurBMNo, NewBMNo : Integer);
```

Description

The method MoveBMNo sets the bookmarknumber of the TBMItem with the number CurBMNo to NewBMNo. Through this, the TBMItem will be moved within the list. If there is another TBMItem with the NewBMNo, that item will be moved correspondingly.

TBMItem Object

Properties Methods

Unit

DBEBMDef

Description

The TBMItem-object is a internal object, used as a container for storing the bookmark defined for a table. It is managed by the TBMManager-object.

Properties

IndexVals
BMNo

Methods

Create
Destroy
GotoMark

Property IndexVals

Used by

TBMItem, Object

Declaration

```
property IndexVals : TStringList;
```

Description

ReadOnly. The property IndexVals offers you access to the index-fieldvalues of the record at the point in time, the bookmark was defined. These values are shown within the Lesezeichendialog and are used for orientation.

Method Create

Used by
TBMManager, Object

Declaration

```
constructor Create(aTbl : TDBDataSet);
```

Description
The constructor Create creates a new manager-object for the referenced table.

Property BMNo

Used by
TBMItem, Object

Declaration
```
property BMNo : Integer;
```

Description
The property BMNo offers you access to the number of the TBMItem. Although you have writing-rights to that property, you must not change it. The value is managed by the owning TBMManager-object.

Method Create

Used by

TBMItem, Object

Declaration

```
constructor create(aTbl : TDBDataSet; BMNo : Integer);
```

Description

The constructor Create creates a new TBMItem-object. A new bookmark is defined on the via aTbl referenced table. The number of the TBMItem is defined by the BMNo-parameter.

Method Destroy

Used by
TBMItem, Object

Declaration
```
destructor Destroy;
```

Description
The destructor Destroy frees the memory, allocated by the object. The bookmark defined on the table is although freed.

Method GotoMark

Used by

TBMItem, Object

Declaration

```
function GotoMark : Boolean;
```

Description

The method GotoMark moves the table to the defined bookmark. The returncode is true, if the record could be found, otherwise false (no exception will be raised).

Method Destroy

Used by

TBMManager, Object

Declaration

```
destructor Destroy;
```

Description

The destructor Destroy frees the memory allocated by the object. All TBMItem-object created through this object are destroyed automatically.

Method MemoView

Used by

TDBExtender, Component

Declaration

```
procedure MemoView(Fld : TField; AsReadOnly : Boolean);
```

Description

This function returns the bookmarkmanager for a given table.

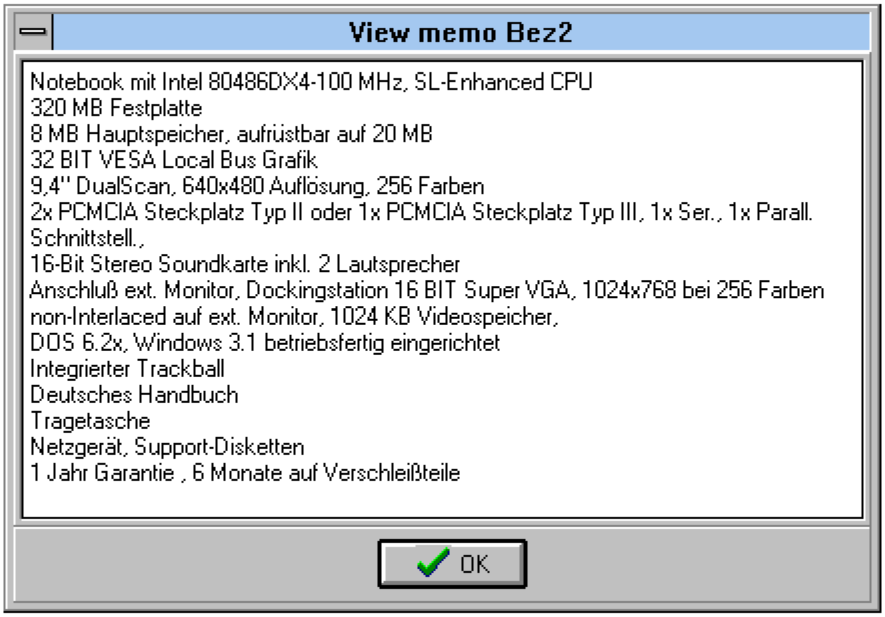
Event OnEditChange

Used by
TDBExtender, Component

Declaration
```
property OnEditChange : TEditChange;
```

Description
This event occurs, if the globaledit-mode is to be changed.

Memoview dialog



Notebook mit Intel 80486DX4-100 MHz, SL-Enhanced CPU
320 MB Festplatte
8 MB Hauptspeicher, aufrüstbar auf 20 MB
32 BIT VESA Local Bus Grafik
9,4" DualScan, 640x480 Auflösung, 256 Farben
2x PCMCIA Steckplatz Typ II oder 1x PCMCIA Steckplatz Typ III, 1x Ser., 1x Parall. Schnittstell.,
16-Bit Stereo Soundkarte inkl. 2 Lautsprecher
Anschluß ext. Monitor, Dockingstation 16 BIT Super VGA, 1024x768 bei 256 Farben non-Interlaced auf ext. Monitor, 1024 KB Videospeicher,
DOS 6.2x, Windows 3.1 betriebsfertig eingerichtet
Integrierter Trackball
Deutsches Handbuch
Tragetasche
Netzgerät, Support-Disketten
1 Jahr Garantie , 6 Monate auf Verschleißteile

This dialog appears, if the user clicks on the speedbutton for memoview or presses F2.
If the memofield can be edited, an OK- and a Cancel-button appear. Inthis case the OK-button will save the modifikations made to the memo. In Readonly-Mode, only a OK-button will be shown..

DBEBMDef Unit

Objects

TBMItem
TBMManager

Constants

MaxCount4NonIdx