Erector Set™ for Data Form (ESDF)

About Erector Set™ for Data Form (ESDF)

Erector Set for Data Form (ESDF) is Rapid Application Development tool for use with of visual database software.

ESDF is a set of mutually aware ActiveX Controls. Each ESDF control encapsulates a specific element of the visual interface along with its function set. To be able to play as a one team ESDF controls need only to be properly linked. Linking is done at design time, there is no need to write glue code.

Because ESDF controls can be linked in a variety of different ways it is possible to build many different visual interfaces. Linking is checked both at design-time and during the initialization phase of run-time.

ESDF controls also provide a visual controlled database structure interface. This means that when you modify the properties of an ESDF control, like Table Name, Field Name, Field Type, Field Size - these changes are automatically reflected in the data base structure. You do not need to go to another shell, to copy data, or modify structures. All of these things are automatically taken care of when you are working in the form designer.

ESDF is competitive with a code-generating tool called Data Form Wizard. Because ESDF is a visual programming tool it is more intuitive and requires fewer manual operations. With ESDF it is much easier to rebuild an already customized application. This is an especially important feature for any Rapid Application Development process.

Comparing ESDF with conventional controls:

A conventional control is an independent component designed to be used in any context. Programming with conventional controls is universal but not very fast because one needs to write glue code to connect several controls.
Whereas, ESDF controls are context dependent. They only work when properly linked together and only for Data Form building. However, programming is more fast.

Technical support

If you have a question about ESDF, first consult online Help, or see Frequently Asked Questions in Custom Software WEB site
http:\\www.rapid-ware.com

If you still don't find the answer, contact us by E-mail:
support@www.rapid-ware.com

Custom Software LLC.
8706 Brightwood Drive,
Fort Washington,
MD 20744
Phone: 301-248-7488
Fax: 301-248-8436

ControlBar Control

See also Properties Events Methods

A ControlBar is an ESDF control you need to be able to manage Tables.

Linking

Every Table has to be linked with a parent ControlBar. A ControlBar can have several child Tables.

Table management

A Table becomes active when its Navigator is focused. The Table becomes inactive when another Table having the same parent ControlBar (belonging the same family) becomes active. So, every moment ControlBar has only one active child Table.

These are actions ControlBar allows to do with an active child Table:

Graphic Control	**Description of action**
Sort ComboBox	Change order of records in the Navigator
Add Button	Go to ADD mode and add new record in the recordset of the Table
Edit Button	Go to EDIT mode and edit current record in the recordset of the Table
Delete Button	Mark current record in the recordset of the Table like deleted
Recall Button	Unmark current record in the recordset of the Table like deleted
Erase Button	Erase from the Table all records marked like deleted
Save Button	Update record in the recordset of the Table with changes
Cancel Button	Roll back changes made in the record
Done Button	Complete edition and go to VIEW mode.

Remarks

A Table control can restrict set of management actions from side of the parent ControlBar. For example, in some of tables records cannot be added directly by user. If some actions are restricted correspondent buttons are invisible on ControlBar. ControlBar refreshes set of graphic controls every time when active table changed.

Properties

CloseVisible

CloseVisible Property

Sets a value indicating whether an Close button in the ControlBar is visible or hidden in run-time.

Syntax

object.CloseVisible [= boolean]

The CloseVisible property syntax has these parts:

Part	**Description**
object	An object expression that evaluates to an ControlBar control
boolean	A Boolean expression specifying whether the object is visible or hidden.

Settings

The settings for boolean are:

Setting	**Description**
True	(Default) Object is visible.
False	Object is hidden.

Remarks

Some of containers don’t Unload method. To hide an Close button at startup, set the Visible property to False at design time.

Note: Most common topics for all Column’s

ColumnAutoID

See also Properties Events Methods

A ColumnAutoID is an ESDF control you can use to insert and maintain Auto ID field in a database. Auto ID field contains a Long value unique for every record. This value generates automatically when record is added and never edited. You may need it for reference purposes.

This control allows have Auto Id field in database of different formats independently of if they support built-in AutoId type of field or not.

Remarks

Typically ColumnAutoID control is invisible in run time.

ColumnNoEdit

See also Properties Events Methods

A ColumnNoEdit control is an ESDF control you can use to display a database field that a user can't change directly.

Remarks

You can write code that modifies value of the ColumnNoEdit control in response to events at run time. For example, if the filed value is dependent of other fields, you need to correct it every time these fields changed.

```
Private Sub MyTable_ValidSave(OK As Boolean)
      nedCharged.DataValue = edtPrice.DataValue*edtAmount.DataValue
End Sub
```

Important

Property DataValue can be changed only when the parent Table is in ADD or EDIT mode. It you are trying to do it in VIEW mode the error generated.

Properties

BrowseColWidth
Browsed
DataField
DataTable
DataValue
Font

Events

BrowseStr
Validate

Applies to

ColumnAutoID
ColumnCode
ColumnEdit
ColumnNoEdit
ColumnSelect

Event Validate

See also Example Applies to

Occurs when a Column control lost focus and its DataValue property is not empty. Column controls are focused only in ADD or EDIT mode. So this event occurs after a user inputs the field and tries to go to another field.

Syntax

Private Sub *ESDFColumn*_Validate ([Index As Integer,] Ok As Boolean)

The Validate event syntax has these parts:

Part	**Description**
ESDFColumn	A placeholder for name of a Column control.
Index	Identifies the control if it's in a control array.
Ok	A Boolean expression specifying whether bound data has changed, as described in Settings.

Settings

The settings for Ok are:

Setting	**Description**
True	Changes accepted
False	Changes don’t accepted and focus comes back

Default value of Ok is True

Remarks

Typically you use this event for checking if input is valid. Like a data type, range and so on.

Example 1

```
‘Returns focus back if not valid
Private Sub edtAmount_Validate (Ok As Boolean)
      If not IsNumeric(edtAmount.DataValue) Then
            Msgbox “Amount must be numeric”
            Ok = False
      End If
End Sub
```

Example 2

```
‘Doesn’t return focus back but modifies field if not valid
Private Sub edtAmount_Validate (Ok As Boolean)
      If not IsNumeric(edtAmount.DataValue) Then
            Msgbox “Amount must be numeric. Correct it later.”
            Amount.DataValue = 0
      End If
End Sub
```

Event BrowseStr

See also Example Applies to

Occurs after the parent Table control requests a Column control for a string value for the Navigator.

Syntax

Private Sub *ESDFcolumn*_ BrowseStr ([Index As Integer,] pstr As String)

The BrowseStr event syntax has these parts:

Part	**Description**
ESDFcolumn	A placeholder for name of a Column control
Index	Identifies the control if it's in a control array.
pstr	A value of the field presented like a string.

Remarks

You can use this event for formatting of columns in Visual Navigator.

For example:

```
pstr = Format(Val(pstr),”0000.00”)
```

Property DataTable

See also Example Applies to

Sets a value that specifies the ESDF Table control is parent for current Column control and through which the current control is bound with a database. Read only at run time.

Remarks

Every Column control must have a parent Table. In run-time ESDF linker tries to link every the Column with the parent Table by using DataTable property. If the linking is impossible error generated.

Data Type

String

Property Font

See also Example Applies to

Returns a Font object.

Syntax

object.Font

The object placeholder represents a Column or Table control.

Remarks

Use the Font property of an object to identify a specific Font object whose properties you want to use. For example, for Table proportional fonts recommended like Courier New.

Property Browsed

See also Example Applies to

Returns or sets a value indicating whether a Column control is presented in the Visual Navigator or not. Read only at run time

Type

Boolean

Property DataField

See also Example Applies to

Returns or sets a value that binds a control to a field in the current record of the parent Table. Read only at run time.

Remarks

Every Column control must have a bound field in the database. In run-time ESDF linker tries to bind every Column with field in the database by using DataField property. If the binding is impossible error generated.

Data Type

String

Property BrowseColWidth

See also Example Applies to

Returns or sets a value indicating number of characters that can be displayed in the Visual Navigator of parent Table control for current Column. Read only at run time.

Remarks

Property Browsed of the Column must be set in True to be able to see the column in the Visual Navigator.

Type:

Integer

Property DataValue

See also Example Applies to

Returns or sets the value of the database bound field.

Syntax

ESDFColumn.DataValue [= value]

The DataValue property syntax has these parts:

Part	**Description**
ESDFColumn	A placeholder for name of a Column control.
String	An expression specifying value of the field.

Remarks

At run time DataValue is read only for all of Column controls but ColumnNoEdit and ColumnEdit.

For ColumnNoEdit and ColumnEdit DataValue can be written when parent table is in ADD or EDIT modes.

Type

Variant

Note: Specific topics for ColumnSelect and ColumnCode

ColumnSelect

See also Properties Events Methods

A ColumnSelect is an ESDF control you can use to display and manage a database field that a user can edit only by selection of alternatives.

Remarks

For properly work ColumnSelect control needs to know:

- Name of directory table in database where alternatives located;
- Name of field from where ColumnSelect takes set of possible alternatives.

ColumnCode

See also Properties Events Methods

A ColumnCode is an ESDF control you can use to display and manage a database field that contains a reference on the record in another database table. Such a field displayed always as a man readable string both in the ColumnCode control and in the Navigator. Not like an ID but like relative to this ID text placed in another table.

A bounded with ColumnCode field user edits by selection of alternatives. Alternatives look for him as man readable strings as well. Differ from ColumnSelect not a string selected is saved after edition but just a reference (ID) on the string.

Remarks
For proper work ColumnSelect control needs to know:

- Name of directory table;
- Name of field in the directory table which contains referred ID;
- Name of field in the directory table which field contains a readable mnemonic.

Properties

BrowseColWidth
Browsed
DataField
DataTable
DataValue
DirDataBase
DirFldString
DirTable
Font

Properties

BrowseColWidth
Browsed
DataField
DataTable
DataValue
DirDataBase
DirTable
DirFldCode
DirFldString
Font

Applies to

ColumnSelect
ColumnCode

Applies to

ColumnCode

Property DirDataBase

See also Example Applies to

Returns or sets the name and location of the source of directory data for a ColumnCode or ColumnSelect control. Read and write at run time.

If your network system supports it, the pathname argument can be a fully qualified network path name such as \\Myserver\Myshare\Database.mdb.
The database type is indicated by the file or directory that pathname points to, as follows:

Pathname Points To...	**Database Type**
.mdb file	Microsoft Access database
Directory containing .dbf file(s)	DBASE database
Directory containing .xls file	Microsoft Excel database
Directory containing .dbf files(s)	FoxPro database
Directory containing .wk1, .wk3, .wk4, or .wks file(s)	Lotus Database
Directory containing .pdx file(s)	Paradox database
Directory containing text format database files	Text format database

For ODBC databases, such as SQL Server or Oracle, this property can be left blank if the control's Connect property identifies a data source name (DSN) that identifies an ODBC data source entry in the registry.
If you change the DatabaseName property after the control's Database object is open, you must use the Refresh method to open the new database.

Property DirTable

See also Example Applies to

Returns or sets the underlying table, SQL statement, or QueryDef object source of directory data for a ColumnCode or ColumnSelect control. Read and write at run time.

Settings

The settings for the property are:

Setting	**Description**
A table name	The name of one of the tables defined in the Database object's TableDefs collection.
An SQL query	A valid SQL string using syntax appropriate for the data source.
A QueryDef	The name of one of the QueryDef objects in the Database object's QueryDefs collection — when accessing a Jet database.

Remarks

The DirTable property specifies the source of the directory records accessible through bound controls on your form.

If you set the DirTable property to the name of an existing table in the database, all of the fields in that table are visible to the bound controls attached to the Data control.

If you set the DirTable property to the name of an existing QueryDef in the database, all fields returned by the QueryDef are visible.

Note At design-time, the QueryDef objects displayed in the Properties window for the DirTable property are filtered out to display only QueryDef objects that are usable with the Data control. QueryDef objects which have parameters, and QueryDef objects which have the following types are not displayed: dbQAction, dbQCrosstab, dbQSQLPassThrough and dbQSetOperation.

If you set the DirTable property to an SQL statement that returns records, all fields returned by the SQL query are visible. If the database you specify in the Database and Connect property isn't a Microsoft Jet engine database, and if the dbSQLPassThrough option is set in the Options property, your SQL query must use the syntax required by that database engine.

Note Whenever your QueryDef or SQL statement returns a value from an expression, the field name of the expression is created automatically by the Microsoft Jet database engine. Generally, the name is Expr1 followed by a three-character number beginning with 000. For example, the first expression would be named: Expr1000.
In most cases you'll want to alias expressions so you know the name of the column to bind to the Column control. See the SQL SELECT statement AS clause for more information.

Property DirFldString

See also Example Applies to

Returns or sets a value that binds a ColumnSelect or ColumnCode control to a field in the directory table which is a source of man-readable alternatives. Read and write at run time.

Remarks

When you use a QueryDef object or SQL statement that returns the results of an expression for DirTable property, the field name is automatically generated by the Microsoft Jet database engine.

Property DirFldCode

See also Example Applies to

Returns or sets a value that binds a ColumnCode control to a field in the directory table which is a referred ID. Read and write at run time.

Note: Specific topics for ColumnEdit

ColumnEdit

See also Properties Events Methods

A ColumnEdit control is an ESDF control you can use to display a database field that a user can change directly in ADD or EDIT mode.

Remarks

You can write also code that changes the field bound with ColumnEdit control in response to events at run time. For example, if the filed value is dependent of other fields, you need to correct it every time these fields changed.

Important

Property DataValue can be changed only when the parent Table is in ADD or EDIT mode. It you try to do it in VIEW mode error generated.

Properties

BrowseColWidth
Browsed
DataField
DataTable
DataValue
Font
KeyColumn

Applies to

ColumnEdit

Property KeyColumn

See also Example Applies to

Returns or sets a value that determines whether an ColumnEdit control manages a key field. Read only at run-time.

Remarks

If KeyColumn is True user can edit the value of key field in ADD mode only, when new record added. He never can edit this field after new record is saved. Often this field needs to be unique. Use the field Validate event to check input.

DataType

Boolean

Example:

```
Dim dns As Recordset, OK as boolean
Set dns = mcntTable.Recordset.Clone
dns.FindFirst DataField + "='" + FieldText.Text + "'"
OK =  Not dns.NoMatch
If Not OK Then
      MsgBox "Value must be unique. Click OK and correct, please.",
vbExclamation
End If
```

Table Control

See also Properties Events Methods

Provides access to data stored in a database and navigation through the data. The Table also provides data for polymorphic collection of Column controls.

Linking

Every the Table has to be linked with one parent ControlBar control which located in the same form. A Table has several columns.   Every column presented by an external Column Control linked with the Table. To be able linked with the Table the Columns are to be in the same form as well.

Graphic Control	**Description of action**
Visual Navigator	Visual Navigator behaves like a List control. Changing the current string in Visual Navigator you also change the current string of the recordset. Column controls display changes automatically.

A Table Control can be in one of these states (modes)

State	**Description**
View	Browsing data in Visual Navigator and Columns.
Add	Cycle of adding of new records. You can add record after record in this state. Entering in the state when clicking button Add on the ControlBar and Exit of state when click the button Done.
Edit	Cycle of editing of existing records. You can edit record after record in this state. Entering in the state when clicking button Edit on the ControlBar and Exit of state when click the button Done.

Behavior of every Column controls depends of state of parent Table.

The Table control is initialized after the initial Form_Load event for the form on which it is placed. If you wish to assign DatabaseName or RecordSource properties in run-time you need to use Form_Load event. You can do enforced initialization the Table control from within Form_Load event using Refresh method. This style allows to perform initialization of the Table before the form becomes visible.

When Visual Basic uses the Jet database engine to create a Recordset, no other Visual Basic operations or events can occur until the operation is complete. However, other Windows-based applications are permitted to continue executing while the Recordset is being created. If the user presses CTRL+BREAK while the Jet engine is building a Recordset, the operation is terminated, a trappable error results, and the Recordset property of the Table control is set to Nothing. In design time, a second CTRL+BREAK causes Visual Basic to display the Debug window.

Properties

ColorActive
ColorInactive
Connect
ControlBar
Database
DatabaseName
EnableAdd
EnableDel
EnableErase
EnableEdit
EnableRecall
Font
Recordset
RecordSource

Events

ControlBarAdd
ControlBarEdit
TableActivated
TableDeactivated
TableReposition
ValidClose
ValidDelete
ValidDone
ValidRecall
ValidSave

Methods
AddOrder
Refresh

Applies to

Table

Event TableActivated

See also Example <u>Applies to</u>

Occurs when Navigator of Table control receives the focus.

Syntax

Private Sub *ESDFTable*_ TableActivated ([Index As Integer])

The TableActivated event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.

Remarks

Typically, you use a TableActivated event procedure to specify the actions dependent of active or inactive state of the Table.

Event TableReposition

See also Example <u>Applies to</u>

Occurs after a record becomes the current record.

Syntax

Private Sub *ESDFTable*_ TableReposition ([Index As Integer]))

The TableReposition event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.

Remarks

Typically, you use a TableReposition event procedure to specify the actions in dependent table. For example to change RecordSource property of driven table to filter records relative with current one.

Note

Occurs after a record becomes the current record.

When a Table control is loaded, the last record in the Recordset object becomes the current record, causing the Reposition event. Whenever a user changes the current record, the Reposition event occurs after each record becomes current.

Event TableDeactivated

See also Example <u>Applies to</u>

Occurs when another table becomes the active.

Syntax

Private Sub *ESDFTable*_ TableDeactivated ([Index As Integer])

The TableDeactivated event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.

Remarks

Typically, you use a TableDeactivated event procedure to specify the actions dependent of active or inactive state of the Table.

Event ControlBarAdd

See also Example Applies to

Occurs when the user presses and then releases a mouse button over a button Add on linked Control Bar control, or presses Enter when the button is focused, or presses combination of keys Alt+A.

Syntax

Private Sub *ESDFTable*_ ControlBarAdd ([Index As Integer])

The ControlBarAdd event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.

Remarks

Typically, you use a ControlBarAdd event procedure to specify the actions

Event ControlBarEdit

See also Example Applies to

Occurs when the user presses and then releases a mouse button over a button Edit on linked Control Bar control, or presses Enter when the button is focused, or presses combination of keys Alt+E.

Syntax

Private Sub *ESDFTable*_ ControlBarEdit ([Index As Integer])

The ControlBarEdit event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.

Remarks

Typically, you use a ControlBarEdit event procedure to specify the actions

Event ValidSave

See also Example Applies to

Occurs when the user presses and then releases a mouse button over a button Save on linked Control Bar control, or presses Enter when the button is focused, or presses combination of keys Alt+S.

Syntax

Private Sub *ESDFTable*_ ValidSave([Index As Integer,]OK As Boolean)
The ValidSave event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.
Ok	Indicates if the action is valid. If value of Ok is FALSE the action will not be performed. Default value is TRUE.

Remarks

Typically, you use a ValidSave event procedure to check record level of data integrity. You can use it also to triggers saving record event, for example to calculate dependent fields.

```
Private Sub MyTable_ValidSave(OK As Boolean)
      If edtPrice.DataValue = ”” OR edtAmount.DataValue =”” Then
            MsgBox ”Input both price and amount before saving”
            OK = False
            Exit Sub
      End If
      nedCharged.DataValue = edtPrice.DataValue * edtAmount.DataValue
End Sub
```

Event ValidRecall

See also Example <u>Applies to</u>

Occurs when the user presses and then releases a mouse button over a button Recall on linked Control Bar control, or presses Enter when the button is focused, or presses combination of keys Alt+R.

Syntax

Private Sub *ESDFTable*_ ValidRecall([Index As Integer,]OK As Boolean)
The ValidRecall event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.
Ok	Indicates if the action is valid. If value of Ok is FALSE the action will not be performed. Default value is TRUE.

Remarks

Typically, you use a ValidRecall event procedure to check reference integrity before recall a record.

Event ValidDelete

See also Example Applies to

Occurs when the user presses and then releases a mouse button over a button Delete on linked Control Bar control, or presses Enter when the button is focused, or presses combination of keys Alt+D.

Syntax

Private Sub *ESDFTable*_ ValidDelete([Index As Integer,]OK As Boolean)

The ValidDelete event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.
Ok	Indicates if the action is valid. If value of Ok is FALSE the action will not be performed. Default value is TRUE.

Remarks

Typically, you use a ValidDelete event procedure to check reference integrity before to mark record as deleted.

Event ValidClose

See also Example <u>Applies to</u>

Occurs when the user presses and then releases a mouse button over a button Close on linked Control Bar control, or presses Enter when the button is focused, or presses combination of keys Alt+C.

Syntax

Private Sub *ESDFTable*_ ValidClose([Index As Integer,]OK As Boolean)

The ValidClose event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.
Ok	Indicates if the action is valid. If value of Ok is FALSE the action will not be performed. Default value is TRUE.

Remarks

This event is typically used to check data integrity on database level.

Event ValidDone

See also Example Applies to

Occurs when the user presses and then releases a mouse button over a button Done on linked Control Bar control, or presses Enter when the button is focused, or presses combination of keys Alt+D.

Syntax

Private Sub *ESDFTable*_ ValidDone([Index As Integer])

The ValidDone event syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Index	Identifies the control if it's in a control array.
Ok	Indicates if the action is valid. If value of Ok is FALSE the action will not be performed. Default value is TRUE.

Remarks

Typically, you use a ValidDone event to trigger a going from Edit or Add mode to View. You can use it also check the table level data integrity.

Method AddOrder

See also Example Applies to

Adds an item to the Order ComboBox of ESDF ControlBar control.

Syntax

ESDFTable.AddOrder pstrMnemonic, pstrCriterion

The AddItem method syntax has these parts:

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
pstrMnemonic	A string will be displayed in Order ComboBox of the parent ControlBar
pstrCriterion	A string will be included in SQL request after ORDER BY

Remark

Typically you use AddOrder method in Form_Load event to define possible sorting orders for the Visual Navigator.

Important

Method 'AddOrder' cannot be used in ADD or EDIT mode otherwise an error generated.

Method Refresh

See also Example <u>Applies to</u>

Forces a complete recreating of the Table’s recordset and repaint of the Navigator.

Syntax

ESDFTable.Refresh

Remarks

Use the Refresh method after you have changed DataBaseName or Recordset properties so changes can take effect. For example for filtering:

```
MyTable.Recordset = “SELECT * FROM Orders WHERE Customer_ID = ” _
+ txtID.Text
MyTable.Refresh
```

Important
Method 'Refresh' cannot be used in ADD or EDIT mode otherwise an error generated.

Property Font

See also Example Applies to
Returns a Font of a control.

Syntax

object.Font

The object placeholder represents an object expression that evaluates to an object in the Applies To list.

Remarks

Use the Font property of an object to identify a specific Font object whose properties you want to use.

Property ControlBar

See also Example Applies to

Sets a value that specifies the ControlBar control is parent for current Table control and through which the current control is managed. Not available at run time.

Remarks

Every Table control must have a parent ControlBar. In run-time ESDF linker tries to link every Table with the parent ControlBar by using ControlBar property of the Table. If the linking is impossible error generated.

Data Type

String

Property Connect

See also Example Applies to

Sets or returns a value that provides information about the source of an open database.

Syntax

ESDFTable.Connect = databasetype;parameters;

The Connect property syntax has these parts.

Part	**Description**
ESDFTable	A placeholder for name of a Table control.
Databasetype	Optional. A String that specifies a database type. For Microsoft Jet databases, exclude this argument; if you specify parameters, use a semicolon (;) as a placeholder.
Parameters	Optional. A String that specifies additional parameters to pass to ODBC or installable ISAM drivers. Use semicolons to separate parameters.

Settings

The Connect property setting is a String composed of a database type specifier and zero or more parameters separated by semicolons. The Connect property passes additional information to ODBC and certain ISAM drivers as needed.

Property RecordSource

See also Example Applies to

Returns or sets the underlying table, SQL statement, or QueryDef object for a Table control.

Syntax

ESDFTable.RecordSource [= value]

The RecordSource property syntax has these parts:

Part	**Description**
ESDFTable	An object expression that evaluates to an object in the Applies To list.
value	A string expression specifying a name, as described in Settings.

Settings

The settings for value are:

Setting	**Description**
A table name	The name of one of the tables defined in the Database object's TableDefs collection.
A SQL query	A valid SQL string using syntax appropriate for the data source.
A QueryDef	The name of one of the QueryDef objects in the Database object's QueryDefs collection — when accessing a Jet database

.

Remarks

The RecordSource property specifies the source of the records accessible through child Column controls on your form.

If you set the RecordSource property to the name of an existing table in the database, all of the fields in that table are visible to the child Column controls linked to the Table control.

If you set the RecordSource property to the name of an existing QueryDef in the database, all fields returned by the QueryDef are visible to the child Column controls linked to the Table control. The order of the records retrieved is set by the QueryDef object's query. For example, the QueryDef may include an ORDER BY clause to change the order of the records returned by the Recordset created by the Table control or a WHERE clause to filter the records. If the QueryDef doesn't specify an order, the data is returned in no particular order.

Note At design-time, the QueryDef objects displayed in the Properties window for the RecordSource property are filtered out to display only QueryDef objects that are usable with the Table control. QueryDef objects which have parameters, and QueryDef objects which have the following types are not displayed: dbQAction, dbQCrosstab, dbQSQLPassThrough and dbQSetOperation.

If you set the RecordSource property to an SQL statement that returns records, all fields returned by the SQL query are visible to the child Column controls linked to the Table control. This statement may include an ORDER BY clause to change the order of the records returned by the Recordset created by the Table control or a WHERE clause to filter the records. If the database you specify in the Database and Connect property isn't a Microsoft Jet engine database, and if the dbSQLPassThrough option is set in the Options property, your SQL query must use the syntax required by that database engine.

Note Whenever your QueryDef or SQL statement returns a value from an expression, the field name of the expression is created automatically by the Microsoft Jet database engine. Generally, the name is Expr1 followed by a three-character number beginning with 000. For example, the first expression would be named: Expr1000.
In most cases you'll want to alias expressions so you know the name of the column to bind to the bound Column control. See the SQL SELECT statement AS clause for more information.

After changing the value of the RecordSource property at run time, you must use the Refresh method to enable the change and rebuild the Recordset.
At run time, if the Recordset specifies an invalid Table name, QueryDef name, or contains invalid SQL syntax, a trappable error will result. If this error occurs during the initial Form_Load procedure, the error is not trappable.

Note Make sure each bound control has a valid setting for its DataField property. If you change the setting of a Table control's RecordSource property and then use Refresh, the Recordset identifies the new object. This may invalidate the DataField settings of child Column controls and cause a trappable error.

Data Type

String

Property EnableAdd

See also Example Applies to

Returns or sets a value that determines whether a Table control allows add new record in the recordset. Read and write at run-time.

DataType

Boolean

Property EnableEdit

See also Example Applies to

Returns or sets a value that determines whether a Table control allows edit current record of the recordset. Read and write at run-time.

DataType

Boolean

Property EnableDel

See also Example Applies to

Returns or sets a value that determines whether a Table control allows mark current record in the recordset like deleted. Read and write at run-time.

DataType

Boolean

Property EnableRecall

See also Example Applies to

Returns or sets a value that determines whether a Table control allows unmark current record in the recordset like deleted. Read and write at run-time.

DataType

Boolean

Property EnableErase

See also Example <u>Applies to</u>

Returns or sets a value that determines whether a Table control allows erase records in the recordset which marked like deleted. Read and write at run-time.

DataType

Boolean

Property DatabaseName

See also Example Applies to

Returns or sets the name and location of the source of data for a Table control. Read and write at run-time.

If your network system supports it, the pathname argument can be a fully qualified network path name such as \\Myserver\Myshare\Database.mdb.
The database type is indicated by the file or directory that pathname points to, as follows:

Pathname Points To...	**Database Type**
.mdb file	Microsoft Access database
Directory containing .dbf file(s)	DBASE database
Directory containing .xls file	Microsoft Excel database
Directory containing .dbf files(s)	FoxPro database
Directory containing .wk1, .wk3, .wk4, or .wks file(s)	Lotus Database
Directory containing .pdx file(s)	Paradox database
Directory containing text format database files	Text format database

For ODBC databases, such as SQL Server or Oracle, this property can be left blank if the control's Connect property identifies a data source name (DSN) that identifies an ODBC data source entry in the registry.
If you change the DatabaseName property after the control's Database object is open, you must use the Refresh method to open the new database.

Property Recordset

See also Example Applies to

Not available at design time. Read/only at run time.

Data Type

Dynaset

Properties ColorActive, ColorInactive

See also Example Applies to

Returns or sets the foreground color used to display text in the Navigator of the Table control in active or inactive states.

Syntax

ESDFTable.ColorActive [= color]
ESDFTable.ColorInActive [= color]

The ColorActive and ColorInActive property syntaxes have these parts:

Part	**Description**
ESDFTable	An object expression that evaluates to an object in the Applies To list.
color	A value or constant that determines the foreground colors of the Table visual navigator, as described in Settings.

Settings

Visual Basic uses the Microsoft Windows operating environment red-green-blue (RGB) color scheme. The settings for color are:

Setting	**Description**
Normal RGB colors	Colors specified by using the Color palette or by using the RGB or QBColor functions in code.
System default colors	Colors specified by system color constants listed in the Visual Basic (VB) object library in the Object Browser. The Windows operating environment substitutes the user's choices as specified in the Control Panel settings.

Remarks

The valid range for a normal RGB color is 0 to 16,777,215 (&HFFFFFF). The high byte of a number in this range equals 0; the lower 3 bytes, from least to most significant byte, determine the amount of red, green, and blue, respectively. The red, green, and blue components are each represented by a number between 0 and 255 (&HFF). If the high byte isn't 0, Visual Basic uses the system colors, as defined in the user's Control Panel settings and by constants listed in the Visual Basic (VB) object library in the Object Browser.

Database Property

Returns the Database object that corresponds to this connection

Return Values

The return value is an object variable that represents a Database object.

While Recordsource is not a table, properties changes cannot be automatically reflected in database structure.

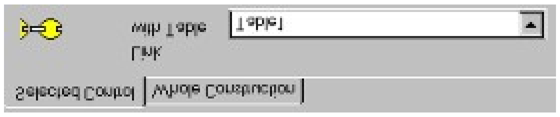
When Recordsource is an SQL querry, you cannot change structure of the Fields manipulating with properties of Column controls.

Errors of initialization:

<ColumnName>: not linked with a Table control

The Column is not an independent control, it cannot work without linking with a Table control. To fix the problem:

1. End your application. Select from Visual Basic menu ‘Run’ then ‘End’.
2. Be sure that the form containing the Column contains a Table control. Table control has image in Toolbox.
3. Open the Page Property of the Column control. For example, click the Column with right mouse button then select ‘Properties’
4. Select a Table from the combo.



5. Click label to check if all the links are correct.
6. Click OK.
7. Close this Help window

<*ColumnName*>: Table <*TableName*> not found

Property DataTable of the Column control doesn’t meet a Table control with the same name in the container form. Perhaps, you have edited name of the Table after the Column control was linked with.

To fix the problem you need re-establish the link. For more about linking see <ColumnName>: not linked with a Table control

<*TableName*>: ControlBar <*ControlBarName*> not found

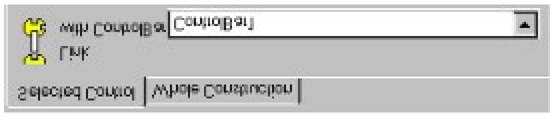
Property ControlBar of the Table control doesn't meet a ControlBar control with the same name in the container form. Perhaps, you have edited name of the ControlBar after the Table was linked with.

To fix the problem you need re-establish the link. For more information about linking see <TableName>: not linked with a ControlBar control

<TableName>: not linked with a ControlBar control

Table is not an independent control, it cannot work without linking with a ControlBar control. To fix the problem:

1. End your application. Select from Visual Basic menu ‘Run’ then ‘End’.
2. Be sure that the form containing <*TableName*> contains a ControlBar control as well. ControlBar control has image in Toolbox.
3. Open the Page Property of <*TableName*> control. For example, click <*TableName*> with right mouse button then select ‘Properties’
4. Select a ControlBar from the combo.



5. Click Whole Construction label to check if all the links are correct.
6. Click OK.
7. Close this Help window

<ColumnName>: Data Field *<FieldName>* not found

Property DataField of the Column control doesn't meet a field with the same name in the database. It can be caused by:

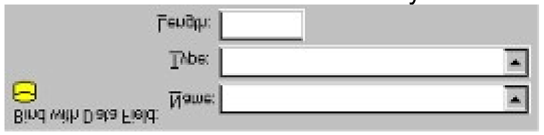
- You have changed structure of the database after the Column control binding
- DatabaseName or Recordsource were re-defined in run-time and structures of design-time database and run-time one are different.
- There was not design-time bound database and name of the field misspelled.

For more about the binding see: <ColumnName>: not bound with a Data Field

<ColumnName>: not bound with a Data Field

Every Column control must be bound with data. When you end your application after the error (menu ‘Run’ then ‘End’), there are two ways to bind the control:

The parent Table control is bound with the data in design-time. To bind the Column:

1. Open the Page Property of <*ColumnName*> control. For example, click <*ColumnName*> with right mouse button then select ‘Properties’
2. Select from Name combo the field you wish to bind with the control.
   

   If the field is absent in the list (correspondingly - in the database), you can add the new field in the database. Just type the name in the combo, select type and type size if the selected type allows it.
3. Click OK.
4. Close this Help window

If by some reason the parent Table can not be bound with the data in design-time the Type and Length combos will be invisible, the Name combo will have an empty list. In this case you can type name of the field in Name combo.

Both init and runtime:

<TableName>: Database *<DatabaseName>* not found

Property DatabaseName of the Table control doesn't meet a database with the same name in the same location. It can be caused by:

- You have changed the database location after the controls was bound with data.
- You have incorrectly re-defined Database property in run-time.

See DatabaseName not defined for more about the Recordsourse defining.

<*TableName*>: DatabaseName not defined

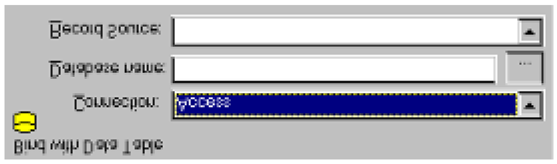
Every Table control need to be bound with data. It can be done both in design-time and in run-time.

It is strictly recommended you use design-time binding. Binding in design time allows you to save time working with the Column controls. Even if design-time a real database is not accessible you can use a ‘proxy’ database with the same structure. In run-time you can re-define the location of database with event Form_Load.

If you create database simultaneously with the application, design-time binding allows you to create and modify the database directly from the designer.

To define a database:

1. Open the Page Property of <*TableName*> control. For example, click the control with right mouse button then select ‘Properties’
2. Select Access from the Connection combo



Record Source:

Database name:

Connection: Access

Bind with Data Table

3. Click the button of Database name label to open the dialog box.
4. Select an existing database or input a new name if you wish to create a new database.
5. Select a table from the Record Source combo or input new name if you wish to create the table. You can also input an SQL querry in the Record Source combo.
6. Click OK
7. Close this Help window

<TableName>: Recordsource *<RecordSourceString>* not found

Property Recordsourse of the Table control doesn’t meet a table of the database with the same name. It can be caused by:

- You have changed the database structure after the controls was bound with data.
- You have incorrectly re-defined Database or Recordsourse property in run-time.

See Recordsource not defined for more about the Recordsourse defining.

<*TableName*>: Recordsource not defined

Recordsource property of a Table control must be defined or in design-time or in initialization of run-time.

To define Recordsource in design time:
1. Open the Page Property of the Table control. For example, click the control with right mouse button then select ‘Properties’
2. Select a table from the Record Source combo or type a new name if you wish to create the table. You can also input an SQL querry in the Record Source combo.

3. Click OK
4. Close this Help window

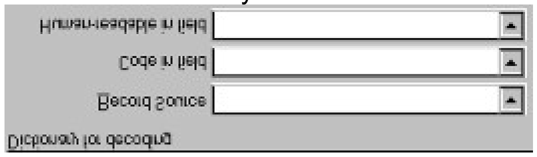
Example of (re)defining of Recordsource in design time:

```
Private Sub Form_Load()
    Table1.Recordsourse = "MyTable"
End Sub
```

<ColumnCodeName>: Error binding with the Directory table

ColumnCode must be bound with a Directory data in design-time.

A directory table must contain at least 2 fields. These fields contain:
- Human-readable form of an record.
- Numeric code of the human-readable form. The code considered to be unique for every record. Your working table contains just a code. ColumnCode allows to display and edit it in man-readable form, using auxiliary Directory table. It considered to be located in the same database as working table.

To bind the control with Directory data Recordsource in design time:
1. Open the Page Property of the Table control. For example, click the control with right mouse button then select ‘Properties’
2. Select a Dictionary table from the Record Source combo, then select the code and human-readable fields in the Directory Table.


3. Click OK
4. Close this Help window

If you re-define Database of the Table control in run-time, check also run-time database contains a Directory table with the same structure as design-time one.

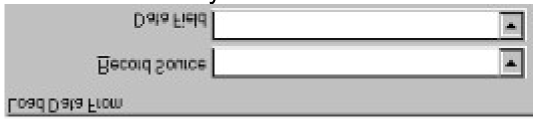
<ColumnSelectName>: Error binding with the Directory table

ColumnSelect must be bound with a Directory data in design-time.

A directory table must contain at least one field from where the ColumnSelect could take alternatives where editing the fields in working table.

To bind the control with Directory data Recordsource in design time:

1. Open the Page Property of the Table control. For example, click the control with right mouse button then select ‘Properties’
2. Select a Dictionary table from the Record Source combo, then select the field in the Directory Table.



3. Click OK
4. Close this Help window

If you re-define Database of the Table control in run-time, check also run-time database contains a Directory table with the same structure as design-time one.

<*TableName*>: Error in SQL <*SQL_String*>

Errors of run-time:

Method 'AddOrder' cannot be used in ADD or EDIT mode

Method 'Refresh' cannot be used in ADD or EDIT mode

DataValue can be assigned in Add or Edit mode only

Internal Error

Options of Selected Control tab (Property Pages of Table control)

Link with ControlBar
Sets property ControlBar of the Table. Table is not an independent control, it cannot work without linking with a ControlBar control. Selecting the name one of accessible ControlBar controls you link the Table with.

Behavior of Table:

Enable Add Record	Sets EnableAdd property. Ability to add a new record in the Table with the ControlBar Add button
Enable Edit Record	Sets EnableEdit property. Ability to edit a record in the Table with the ControlBar Edit button
Enable Delete Record	Sets EnableDel property. Ability to mark a record like deleted in the Table with the ControlBar Delete button
Enable Recall Record	Sets EnableRecall property. Ability to un-mark a record like deleted in the Table with the ControlBar Recall button
Enable Erase Record	Sets EnableErase property. Ability to unmark a record like deleted in the Table with the ControlBar Erase button

Bind with Data Table:

Connection	Sets Connect property of the Table control. Select type of connection.
Database name	Sets DatabaseName property of the Table control. Click ... button to select with the dialog existing database or type in the dialog name of new database you wish to create.
Record Source	Sets RecordSource property of the Table control. Select the name of a data Table from the combo or type a new name if you wish to create a new data Table.

Options of Selected Control tab (Property Pages of ColumnEdit, ColumnNoEdit, ColumnAutoId controls)

Link with Table
Sets DataTable property of the Column control. The Column is not an independent control, it cannot work without linking with a Table control. Selecting the name one of accessible Table controls you link the Column control with.

Behavior in the Table:

Key Column	Key field is an identifier of a record than cannot be changed during living cycle of the record. Typically it is used for referencing of the record. Key Columns cannot be edited in Edit mode. Only in Add mode when adding a new record. Key can be both unique and not unique. If you need an unique Key you can insert the code in Validate event.
Browsed	Sets Browsed property. If the Column is visible in the Visual Navigator of the parent Table.
Width	Sets BrowseColWidth property. Width of visible column in the Visual Navigator of the Table.

Bind with Data Field:

Name	Sets DataField property. Name of the field in the database that is bound with the Column control. Select the name from the combo or type a new name if you wish to add the new field in the database.
Type	Type of the field in the database that is bound with the Column control. If you have selected the existing field, type is set automatically. However, you can change the type of the field in any time. It doesn’t matter if the data Table is empty or contains data, ESDF will change the type of field in the database saving data if it is possible.
Length	Length of the field in the database Column control is bound with. This option is visible only for appropriate Type. You can set or change the Length in any time. Structure of the database will be changed correspondingly.

Options of Selected Control tab (Property Pages of ColumnSelect control)

Link with Table
Sets DataTable property of the Column control. The Column is not an independent control, it cannot work without linking with a Table control. Selecting the name one of accessible Table controls you link the Column control with.

Behavior in the Table:

Browsed	Sets Browsed property. If the Column is visible in the Visual Navigator of the parent Table.
Width	Sets BrowseColWidth property. Width of visible column in the Visual Navigator of the Table.

Bind with Data Field:
Sets DataField property. Name of the field in the database that is bound with the Column control. Select the name from the combo or type a new name if you wish to add the new field in the database. Type and Length are not important here because they are defined by the field that is the source of the data.

Load data from:

Record Source	Select from the combo name of the data Table which will be the source of alternatives for the ColumnSelect when adding or editing.
Data Field	Select from the combo name of the field which will be the source of alternatives for the ColumnSelect when adding or editing.

Options of Selected Control tab (Property Pages of ColumnCode control)

Link with Table
Sets DataTable property of the Column control. The Column is not an independent control, it cannot work without linking with a Table control. Selecting the name one of accessible Table controls you link the Column control with.

Behavior in the Table:

Browsed	Sets Browsed property. If the Column is visible in the Visual Navigator of the parent Table.
Width	Sets BrowseColWidth property. Width of visible column in the Visual Navigator of the Table.

Bind with Data Field (numeric only):

Coded field
Sets DataField property. Name of the field in the database that is bound with the Column control. Select the name from the combo or type a new name if you wish to add the new field in the database. Type and Length are not important here because are defined by the field that is the source of the data.

Dictionary for decoding:

Record Source	Sets DirTable property. Select from the combo name of the Directory Table which will be the source of man-readable records for the ColumnCode control.
Numeric ID in Field	Sets DirFldCode property. Field containing numeric ID of a human-readable record. The field bound with the ColumnCode contains this ID like a code of the human-readable form.
Human-readable in field	Sets DirTable property.

Options of Whole Construction tab (Property Pages of all controls)

Linked controls
Hierarchy list of properly linked controls.

Free controls
List of not linked controls.