DBImport ActiveX V 1.5

Properties Methods Events Technical Support

Description

The ActiveX DBImport enables you to import any file in a professional way.
You can define settings using a wizard, choosing the type of file you import (variable, fixed, access, paradox, odbc...) the separator character, the fields you want, the number of records you want, and many other setttings.

The mandatory properties are the DatabaseName and TableName (the table you want to import to), and the IniFile (the file that will hold the defined settings).After initializing these properties (either using the properties window or directly in your code), you can call either the method execute to allow the user import the tables interactively; or call the method BatchImport to do batch importing.

Use the property ImportCaption if you want to change the default caption, the property ImportFile enables you to initialize the name of the import file, if you provide a help file set the HelpId to point to the help topic

ConnectString
DataBaseName
DSN
HelpId
ImportCaption
ImportFile
IniFile
MultiTable
PassWord
QuitAfterImport
ShowProgress
ShowSummary
ShowErrors
TableName
UserName

Technical Support

You can contact us directly through these :

Internet : support@lvdisoft.com
Compuserve :100733,1402 or lvdi
Web site : http://www.lvdisoft.com
Tel : +33 6 07 89 26 57
Fax : +33 1 46 63 42 31
Address : LVDI Software 67, rue Charles Frerot 94250 Paris-Gentilly France

Property DataBaseName

Applies to

DBImport component

Syntax

object.DataBaseName [= string]

The DataBaseName property syntax has these parts:

Part	Description
string	*A string expression that evaluates to the name of a Database.*

Description
Set this property to the name of the database that contains the table to which data will be imported.

Example

```
DBImportDialog1.DataBaseName = "c:\test\MyData.mdb"
DBImportDialog1.IniFile = "iSettings.ini"
DBImportDialog1.execute
```

Property ImportCaption

Applies to

DBImport component

Syntax

object.Caption [= string]

The Caption property syntax has these parts:

Part	**Description**
string	*A string expression that evaluates to the text displayed as the caption.*

Default
"Import Data"

Description
Set this property to display a caption (window title) that suites your needs.

Example

```
DBImportDialog1.ImportCaption = "Import customer data"
```

Property IniFile

Applies to

TDBImport component

Syntax

object.IniFile [= string]

The IniFile property syntax has these parts:

Part **Description**
string *A string expression that evaluates to the name of a parameter file that will hold user settings.*

Description
Set this property to the name of the IniFile that will contain the settings that the user will save.
Don't forget that you prepare a setting for a known table, this setting will contain field names, if you try to apply that setting to another table (that have different field names) you will get an error message :'This setting is incompatible with the current table'.

You can have one inifile storing settings for different tables if you give a significant name for each setting, like 'CustomerSetting' for example.

Example

```
DBImportDialog1.DataBaseName = "c:\test\MyTable.mdb"
DBImportDialog1.IniFile = "Settings.ini"
DBImportDialog1.execute
```

Property ImportFile

Applies to

DBImport component

Syntax

object.ImportFile [= string]

The ImportFile property syntax has these parts:

Part	Description
string	*A string expression that evaluates to the default intput file name.*

Default
""

Description
Set this property to initialize the edit box file name. If you don't set this property the edit box will be empty (if it's the first import) or will hold the last used filename.

Example

```
DBImportDialog1.ImportFile = "c:\tmp|import.txt"
```

Property HelpId

Applies to

DBImport component

Syntax

object.HelpId [= integer]

The HelpId property syntax has these parts:

Part **Description**
integer *A numeric expression specifying the Help Id to use.*

Default
0

Description
If you provide a help file you must set this property to point to your help file topic.number. Leave it to zero if you don't provide a help file.

Example

```
DBImportDialog1.HelpId = 133
```

Property QuitAfterImport

Property QuitAfterImport

Applies to

DBImport component

Syntax
object.Enabled [= boolean]

The Enabled property syntax has these parts:

Part	**Description**
boolean	*A Boolean expression that specifies whether the DBImport dialog is closed after the import is finished.*

Default
False

Description
This property enables you to decide weather the wizard dialog will close itself after finishing the import. By default it is set to 'False', i.e the dialog will stay until the user clicks on the button cancel.

Example

```
DBImportDialog1.QuitAfterImport = True
```

Event FilterTable

Applies to

DBImport component

Syntax

Private Sub DBImport1_FilterTable(ByVal TableName As String, Hide As Boolean)

The FilterTable event syntax has these parts:

Part	Description
TableName	A string that correspond to the table name to which data is imported.
Hide	A boolean that indicates if that table will be available for import to the user .

Description

This event happens before a table is proposed to the user for import.
If set to hide, the user will not see that table (as if it were not part of the DataBase)

This event is significant only in the MultiTable case.

Example

```
Private Sub DBImport1_FilterTable(ByVal TableName As String, Hide As Boolean)
    If TableName = "Employees" Then Hide = True
End Sub
```

Event FilterFileType

Applies to

DBImport component

Syntax

Private Sub DBImport1_FilterType(ByVal FileType As Integer, Hide As Boolean)

The FilterType event syntax has these parts:

Part	Description
FilterType	*An integer that represents a file type, can be any of : TVariable, TFixed, THtmlTags, TAccess, TParadox3, TParadox4, TParadox5, TDBase3, TDBase4, TDBase5, TFoxpro20, TFoxpro25, TFoxpro26, TFoxpro30*
Hide	*A boolean that indicates if that filetype will be available for import to the user .*

Description

This event happens before a filetype is proposed to the user as a type for the input file. If set to hide, the user will not see that file type.

Example

```
Private Sub DBImport1_FilterType(ByVal FileType As Integer, Hide As Boolean)
    ' filter out Foxpro 2.0 tables
    If FileType = TFoxpro20 Then Hide = True
End Sub
```

Property DSN

Applies to

DBImport component

Syntax

object.DSN [= string]

The DSN property syntax has these parts:

Part	Description
string	*A string expression that evaluates to the name of an ODBC Datasource.*

Description
Use this property to import to an ODBC DataSource

Example

```
DBImportDialog1.DSN = "BooksDSN"
DBImportDialog1.IniFile = "iSettings.ini"
DBImportDialog1.execute
```

Execute
BatchImport

Method Execute

Applies to

DBImport component

Syntax

object.Execute

Description
This method displays the import dialog box and returns when the user clicks on the cancel button.

Example

```
DBImportDialog1.DataBaseName = "c:\test\MyTable.mdb"
DBImportDialog1.ConnectString = Access
DBImportDialog1.IniFile = "eSettings.ini"
DBImportDialog1.execute
```

Event BeforeFieldImport

Applies to

DBImport component

Syntax
Private Sub DBImport1_BeforeFieldImport(ByVal TableName **As String**, ByVal FieldName **As String**, ByVal FieldType **As Integer**, ByVal FieldSize **As Integer**, bufField **As String**)

The BeforeFieldImport event syntax has these parts:

Part	**Description**
TableName	*A string that correspond to the destination table name*
FieldName	*The field that is currently imported*
FieldType	*The type of the field that is currently imported (can be any of : dbBigInt dbBinary dbBoolean dbByte dbChar dbCurrency dbDate dbDecimal dbDouble dbFloat dbGUID dbInteger dbLong dbMemo dbNumeric dbSingle dbText dbTime dbTimeStamp)*
FieldSize	*Integer representing the size of the field*
bufField	*The value that is going to be imported.*

Description
This event happens before a value is imported, it could be useful in many cases for example formatting a number with leading zeros and a sign.

Use the FieldName, FieldType, and FieldSize informations to choose the field you want to change and use the FieldValue to assign a new value.

Example

```
'In this example we need to insert a comma in a special field
'Input : 50000 output : 500,00

Private Sub DBImport1_BeforeFieldImport(ByVal TableName As String, ByVal FieldName As
String, ByVal FieldType As Integer, ByVal FieldSize As Integer, bufField As String)
        if   FieldName = "Amount"   then
                bufField = Left(bufField,Len(FieldValue)-2) + ","+Right ( bufField,2)

end sub
```

BeforeFieldImport
BeforeRecordImport
ValidateAfterPost
ValidateBeforePost
FilterField
FilterTable
FilterFileType

Event ValidateBeforePost

Applies to

DBImport component

Syntax
Private Sub DBImport1_ValidateBeforePost (RecordCanceled As Boolean, ProcessCanceled As Boolean)

The ValidateBeforePost event syntax has these parts:

Part	Description
RecordCanceled	A boolean that indicates if the current record is canceled or not .
ProcessCanceled	A boolean that indicates if the hole process is canceled or not .

Description
This event happens before a record is posted to the DataBase

Example

```
Private Sub DBImport1_ValidateBeforePost(RecordCanceled As Boolean, ProcessCanceled As Boolean)
        If nbRecords = 100 then ProcessCanceled = TRUE
End Sub
```

Event ValidateAfterPost

Applies to

DBImport component

Syntax
Private Sub DBImport1_ValidateAfterPost (ProcessCanceled As Boolean)

The ValidateAfterPost event syntax has these parts:

Part	Description
ProcessCanceled	A boolean that indicates if the process is canceled or not .

Description
This event happens after a record is posted to the DataBase

Example

```
Private Sub DBImport1_ValidateAfterPost( ProcessCanceled As Boolean)
        If nbRecords = 100 then ProcessCanceled = TRUE
End Sub
```

Property ShowErrors

Applies to

DBImport component

Syntax
object.ShowErrors [= boolean]

Default
True

Description
This property enables you to show or hide the errors that might occur during the import process.
This flag can be useful when doing batch import, since errors are displayed to the user as dialog boxes, if you don't set this flag to False when doing batch, the import process will be halted if an error occurs.

Example

```
DBImportDialog1.ShowErrors = False;
DBImportDialog1.BatchImport ("customers")
```

Property ShowProgress

Applies to

DBImport component

Syntax
object.ShowProgress: [= boolean]

Default
True

Description
This property enables you to show or hide the progress bar, this can be useful when doing batch processing (no user interaction), and also to speedup the import process.

Example

DBImportDialog1.ShowProgress = False

Property ShowSummary

Applies to

DBImport component

Declaration
object.ShowSummary: [= boolean]

Default
True

Description
When this property is set to True, at the end of the import the component will display how many records were imported
Set this property to False when doing batch imports to avoid blocking the process.

Example

```
DBImportDialog1.ShowSummary = False;
```

Method BatchImport

Applies to

DBImport component

Syntax
object.BatchImport setting

The BatchImport method syntax has these parts:

Part	Description
setting	Required. string expression specifying the name of a setting that was defined in the wizard.

Description
This method allows to do importing in batch mode, without user interface.
A previously prepared setting should be used.
The flags ShowErrors and ShowSummary must set to False to avoid any blocking dialog,
the flag ShowProgress can be True since it's non-blocking, but it slows down the process a little bit.

Example

```
DBImportDialog1.DataBaseName = "c:\temp\nwind.mdb"
DBImportDialog1.TableName = "Customers"

DBImportDialog1.IniFile = "iSettings.ini"
DBImportDialog1.ShowSummary = False
DBImportDialog1.OverWrite = True
DBImportDialog1.ImportFile = "c:\tmp\out.txt" 'do not forget to specify the input file!
DBImportDialog1.BatchImport("CustomersSetting")
```

Event FilterField

Applies to

DBImport component

Syntax

Private Sub DBImportDialog1_FilterField(ByVal TableName As String, ByVal FieldName As String, Hide As Boolean)

The FilterField event syntax has these parts:

Part	Description
TableName	*A string that corresponds to the table name being imported.*
FieldName	*The field that is currently imported*
Hide	*A boolean that indicates if that field will be available for import to the user.*

Description

This event happens before a field is proposed to the user for import.

If set to hide, the user will not see that field (as if it were not a column of the table)

Example

```
Private Sub DBImport1_FilterField(ByVal TableName As String, ByVal FieldName As String, Hide As Boolean)
        If TableName = "Employees" And FieldName = "Salary" Then Hide = True
End Sub
```

Property UserName

Applies to

DBImport component

Syntax

object.UserName [= string]

The UserName property syntax has these parts:

Part	Description
string	*A string expression that evaluates to the name of a user.*

Description

Use this property to import from an ODBC DataSource that requires a Username.

Example

```
DBImportDialog1.DSN = "BooksDSN"
DBImportDialog1.UserName = "scott"
DBImportDialog1.PassWord = "tiger"
DBImportDialog1.execute
```

Property PassWord

Applies to

DBImport component

Syntax

object.PassWord [= string]

The PassWord property syntax has these parts:

Part	Description
string	*A string expression that evaluates to the password of a user.*

Description

Use this property to import from an ODBC DataSource that requires a Username and password.

Example

```
DBImportDialog1.DSN = "BooksDSN"
DBImportDialog1.UserName = "scott"
DBImportDialog1.PassWord = "tiger"
DBImportDialog1.execute
```

Property ConnectString

Applies to

DBImport component

Syntax

object.ConnectString [= TypeConnect]

The ConnectString property syntax has these parts:

Part **Description**
TypeConnect A connect type that indicates what is the source database (that you want to import to), it can be any one of : Access, Paradox3, Paradox4, Paradox5, DBase3, DBase4, DBase5, Foxpro20, Foxpro25, Foxpro26, Foxpro30, ODBC

Default
Access

Description
Set this property to the type of the database to which you want to import data.

Example

```
DBImportDialog1.DataBaseName = "c:\test\MyTable.mdb"
DBImportDialog1.ConnectString = Access
DBImportDialog1.IniFile = "iSettings.ini"
DBImportDialog1.execute
```

Property MultiTable

Applies to

DBImport component

Syntax
object.MultiTable [= boolean]

The MultiTable property syntax has these parts:

Part **Description**
boolean A Boolean expression that specifies whether the DBImport dialog will be allow the user to choose the destination import table.

Default
False

Description
This property enables you to show or hide the first page of the DBImport wizard, which displays the list of tables of the current database, allowing the user to choose which table to import data into .
If you set this property to false, then the property TableName must not be empty.

Example

```
DBImportDialog1.MultiTable = True
```

Property TableName

Applies to

DBImport component

Syntax

object.TableName [= string]

The TableName property syntax has these parts:

Part Description
string A string expression that evaluates to the name of the table to which you want to import data into.

Description
Set this property to the name of the table that you to import.

Example

```
DBImportDialog1.DataBaseName = "c:\test\MyTable.mdb"
DBImportDialog1.TableName = "Customers"
DBImportDialog1.ConnectString = Access
DBImportDialog1.IniFile = "iSettings.ini"

DBImportDialog1.execute
```