TSQLSet v1.03

Properties

Methods

Unit sqlset

Compatible with Delphi 3/4/5/6

Requires the components DList v1.33 and Reader v3.0 or later versions

Description

Component for construction of list of SQL queries, editable in the property editor at design time, with possibility of insertion variables in query, which can be replaced by any string value at run time.

Installation

Copy the following files in your SQLSet directory

- 1) SQLStReg.pas (from folder d3-5 or d6)
- 2) SQLStReg.dcr
- 3) SQLSEdit.pas
- 4) SQLSEdit.dfm
- 5) SQLSet.pas or SQLSet.dcu

This component requires installed components: Dinamic List v1.33 and Text Reader v3.0 or later versions, but if You do not require this components separately, You can only copy the following additional units in your SQLSet directory:

- 1) DList.pas or DList.dcu;
- 2) Reader.pas or Reader.dcu.

Then install component SQLSet -

Component / Install Component / Browse / SQLStReg.pas / Ok / Compile.

Make sure that path of your SQLSet directory is indicated - Tools / Environment Options / Library / Library path.

Properties

<u>SQLItems</u>

Methods

GetSQL

RegisterVar • AddSQL

(

ClearItems
ClearVariables
RemoveSQL

RemoveVar

SQLItems property

See also

Applies to TSQLSet component

Declaration

property SQLItems: TSQLItems;

Description

Property for construction of list of SQL querys at design time.



<u>ClearItems</u> method

<u>ClearVariables</u> method

GetSQL method

Example

Applies to

TSQLSet component

Declaration

function GetSQL(const Title: string): string;

Description

This function returns SQL text, which is contained in list of property SQLItems under the name Title changing variables by run time string values. If such title is absent, the function will return free value. "Title" is not case sensitive.

GetSQL method example

```
Query1.SQL.Text := SQLSet1.GetSQL('Customers');
Query1.Open;
```

RegisterVar method

See also

Example

Applies to

TSQLSet component

Declaration

function RegisterVar(const VarName, Value: string): boolean;

Description

Declares a variable under the name VarName and determines it Value. If the variable under this name is already declared, to it only will be changed value.

These variables will changed by its values.

If VarName contains inadmissiblis characters, the variable will not be registered, and function will return false.

Inadmissiblis characters are all delimiters of SQL and spaces.

"VarName" is not case sensitive.



ClearVariables methodRemoveVar method

RegisterVar method example

```
if SQLSet1.RegisterVar('CountryVar', ComboBox1.Text) then begin
  Query1.SQL.Text:=SQLSet1.GetSQL('Customers');
  Query1.Open;
end;
```

AddSQL method

See also

Applies to

TSQLSet component

Declaration

procedure AddSQL(const Title, SQLText: string);

Description

Adds to gang a new SQLText under the name Title at run time. "Title" is not case sensitive.



ClearItems method

See also

Applies to

TSQLSet component

Declaration
procedure ClearItems;

DescriptionClears list of SQL querys at run time.



ClearVariables method

See also Applies to TSQLSet component

Declaration procedure ClearVariables;

DescriptionClears all declared variables at run time.



RemoveSQL method

See also

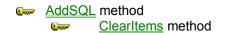
Applies to TSQLSet component

Declaration

procedure RemoveSQL(const Title: string);

Deletes from a list SQL query under the name Title at run time.

"Title" is not case sensitive.



RemoveVar method

See also

Applies to

TSQLSet component

Declaration

procedure RemoveVar(const VarName: string);

Description

Deletes declared variable under the name VarName at run time.

"VarName" is not case sensitive.

