

TSQLBuilder

Version 1.40

Delphi Choice Award 1996 Winner



TSQLBuilder wins "Delphi Choice '96 Award".

If you would like more information about the "Delphi Choice Award"

you can get it from The Delphi Information Connection at

<http://www.delphi32.com/apps/dchoice.html>

TSQLBuilder is a non visual component for Delphi. It only takes one line of code to allow your application to visually build, run and view the results of multi-table SQL queries for any database tables in any alias. Select queries may be run and the result table viewed all within the component. It even includes an expression builder to make your life even easier. Use it either as a standalone query tool or just read off the properties and run the query yourself for total control. Queries can be saved to or loaded from BlobStreams so they can be stored in database blob fields. Additionally they may be saved to or loaded from disk either automatically or manually.

How to Use TSQLBuilder

TSQLBuilder is used by the programmer in much the same way as the standard "OpenFile" or "SaveFile" dialogs.

You set the components properties as required and then call the execute function, everything from then until the user clicks the Ok or Cancel button is handled by the component. If the Ok button is clicked then execute returns *true* and the SQL property contains the query.

If the user has fine-tuned the query by hand in the Query Editor panel then the SQLModified property is set to True, the modified SQL query is stored in the UserSQL property. The original "built" query is still stored in the SQL property, it's up to you whether you want to use the original query or the modified query or simply use SQLBuilder on it's own and simply ignore both.

A sample application demosqlb.dpr is included in this package. For a description of how the user actually generates queries using the component see sqlbuild.wri.

Published Properties

Alias : The default alias that TSQLBuilder will use when executed and if changing the alias is allowed it will be updated to contain the last chosen alias.

Options

sbAllowUpdates : Allow generation of Update queries.

sbAllowDeletes : Allow generation of Delete queries.

sbAllowChange : Allow user to change the current alias.

sbShowSystemTables: Show database system tables for sql databases only.

sbEnableEditor :Allow modifying of query text in sql text panel.

QueryType : The default QueryType that TSQLBuilder will use when executed. Possible QueryType values are SelectQuery, UpdateQuery, DeleteQuery.

SQLType : This determines which version sql syntax will be generated. The default is SQLAuto other possible values are SQL92, SQL89 and LocalSQL. SQLAuto means that SQLBuilder will attempt to use SQL92 syntax for server based sql (except for Informix for which it will use SQL89) and LocalSQL for Paradox, Dbase and Heterogeneous queries.

PrivateDir : Directory SQLBuilder can use to store temporary files. If empty the the applications directory is used.

AutoloadLastQuery : If true and a SQLBuilder stream file exists in the PrivateDir then the query stream will be loaded.

QueryFile: File name of a previously saved stream file that TSqlbuilder will load when executed

Public

Methods

function Execute:Boolean;

Execute is a function which returns a boolean value. It is *true* if the user clicked the OK button else it is *false*. Call this function to display TSQLBuilder to build the query.

procedure StoreQueryToStream(AStream:TStream);

Stores the query in any open Tstream descendent.

procedure StoreQueryToFile(fName:TFilename);

Stores the query to the given pathname.

procedure LoadQueryFromStream(AStream:TStream);

Load the query form an open stream.

Properties

Property SQL :TStrings

Contains the generated query when user clicks OK button

Property SQLModified :Boolean

Returns true if the user has used the query text window to modify the query.

Property UserSQL :TStrings

Contains the user modified query when user clicks OK button

Property IsMultiAlias :Boolean

Returns true if more than one alias is involved in the query.

Property UsedAliasList :TStrings

Returns a list of all databases/aliases used in the query.

Property IsHeterogeneous:Boolean

Returns true if more than one database type is used in the query

Property IsAllStandard:Boolean

Returns true if all database types used in the query are "STANDARD" database types

Property LocalSqlNeeded:Boolean

Returns true if the query is heterogeneous or if it only uses "STANDARD" type databases.

Property AutoGroup:Boolean

If true the Grouping of non-summary fields in queries with a summary field is enforced. This setting is mirrored in the main form's "Autogroup" checkbox.

Property UseWizardOnShow:Boolean

Displays the Query Table and Field Wizard on show.

Evaluation Package

This package contains the evaluation version of TSQLBuilder and should include the following files:

readme.wri (this file)

regform.wri (registration form)

file_id.diz (brief description of this package)

BTNEDIT.ZIP, SQLBLDD1.zip and/or SQLBLDD2.zip (component)

Please redistribute this package exactly as you received it, without adding or removing anything.

BTNEDIT.ZIP contains a simple component that is basically an TEdit component with a button and a OnClick event. It is used as an in-cell editor by the TSQLBuilder component, but it is a handy little component on it's own. It is freeware and source is available on request.

SQLBLDD1.zip contains the Delphi 1.x version with the following files;

sqlbreg.dcu *Component registration unit for BtnEdit and TSqlBuilder.*

sqlbuild.dcu *Delphi unit containing TSQLBuilder component*

sqlbuild.dcr *dcr resource file for TSQLBuilder*

sqlbldD1.res *res resource file for TSQLBuilder*

sqlbld16.res *res resource file for TSQLBuilder*

sqlbld?? .dcu *Support units*

sqlbld?? .dfm *Support units*

demosqlb.dpr *project file for the Demonstration of using TSQLBuilder*

demomain.pas *main file for the TSQLBuilder demo*

demomain.dfm *form resource file for the Demonstration of using TSQLBuilder*

EmpOrds1.stm : *Sample query using DBDEMOS alias*

EmpOrds2.stm : *Sample query using DBDEMOS alias*

CustOrd1.stm : *Sample query using DBDEMOS alias*

Hetqry1.stm : *Sample query using IBLOCAL and DBDEMOS alias*

Note: sqlbld04.pas and sqlbld08.pas do not have corresponding *.dfm files.

SQLBLDD2.zip contains the Delphi 2.0 version with the following files;

sqlbreg.dcu *Component registration unit for BtnEdit and TSqlBuilder.*
sqlbuild.dcu *Delphi unit containing TSQLBuilder component*
sqlbldD2.res *res resource file for TSQLBuilder*
sqlbld32.res *res resource file for TSQLBuilder*
sqlbld??.dcu *Support units*
sqlbld??.dfm *Support units*

demoqlb.dpr : *project file for the Demonstration of using TSQLBuilder*
demomain.pas : *main file for the Demonstration of using TSQLBuilder*
demomain.dfm : *form resource file for the Demonstration of using TSQLBuilder*
EmpOrds1.stm : *Sample query using DBDEMOS alias*
EmpOrds2.stm : *Sample query using DBDEMOS alias*
CustOrd1.stm : *Sample query using DBDEMOS alias*
Hetqry1.stm : *Sample query using IBLOCAL and DBDEMOS alias*

Note: sqlbld04.pas and sqlbld08.pas do not have corresponding *.dfm files.

Installation

This package contains the evaluation version of the TSQLBuilder component for Borland Delphi. Generally this zip file will contain two zip files one for each of the separate versions of the component for Delphi 1 and Delphi 2 and another for the free BtnEdit component, however some sites forbid uploading the 2 versions together (e.g. Compuserve's Delphi forum) and in that case the two zip files to download are SQLBLDD1.ZIP (Delphi 1) and SQLBLDD2.ZIP (Delphi 2).

Important: Make a backup copy of complib.dcl (delphi 1.x) or cmplib32.dcl (Delphi2.0), if anything should go wrong the simply exit Delphi and restore the backup.

Remove any old versions of TSQLBuilder from the library and rebuild the library.

Installation Delphi 1.x

Unzip sqlbldD1.zip file in some temporary directory and copy following files;
Note: Use pkunzip with the -d parameter to preserve the directory structure when unzipping SQLBLDD1.ZIP.

Copy the files from the COMP sub-directory

sqlbreg.dcu
sqlbuild.dcr
sqlbuild.dcu
sqlbuild.res
sqlbldD1.res
sqlbld16.res
all sqlbld??.dcu
all sqlbld??.dfm

and all the files in BtnEdit.Zip
to delphi\lib directory.

Copy following files from the Demos sub-directory

demo*.* {Demo program source code}
*.stm {Example queries}

to the directory of your choice.

Then run Delphi and install sqlbreg.dcu into your library. TSQLBuilder and TBtnEdit will appear on the samples page of component palette. After this you can load and compile demosqlb.dpr.

Installation Delphi 2.0

Unzip sqlbldD2.zip file in some temporary directory and copy following files;
Note: Use pkunzip with the -d parameter to preserve the directory structure when unzipping SQLBLDD1.ZIP.

Copy the files from the COMP sub-directory

sqlbreg.dcu
sqlbuild.dcr
sqlbuild.dcu
sqlbuild.res
sqlbldD2.res
sqlbld32.res
all sqlbld??.dcu
all sqlbld??.dfm

and all the files in BtnEdit.Zip
to your delphi component directory.

Copy following files from the Demos sub-directory

demo*. * {Demo program source code}
*.stm {Example queries}

to another directory of your choice.

Then run Delphi and install sqlbreg.dcu into your library. TSQLBuilder and TBtnEdit will appear on the samples page of component palette. After this you can load and compile demosqlb.dpr.

What do I get if I register?

Unlimited tables in a query.
Unlimited User defined fields.
Both 16bit & 32 bit components
Developers Property Editor version of TSQLBuilder.
Free upgrade to next version.
Support from the author.
Optional full source code for both SQLBuilder components and property editors.
The nice warm feeling inside that comes from doing the right thing.

Registration

TSQLBuilder is shareware. You are licensed to evaluate the software for 30 days, registration is US\$45.00 and includes full source code. When I receive notification of you registration I'll send you the registered version of the package via Email (usually within 24 hours). Australian users should send a email or fax the author for local pricing.

How To register

Internet: You may register TSQLBuilder live on our www page, the address is;
<http://ourworld.compuserve.com/homepages/sharewar.htm>

Compuserve: GO SWREG and register by selecting registration number 11535.
Note: Using SWREG will automatically select the source code option.

Telephone: CREDIT CARD ORDERS ONLY - Quote ITEM-14679
"You can order with MC, Visa, Amex, or Discover from Public (software)
Library by calling 800-2424-PsL or 713-524-6394 or by FAX to 713-524-6398
or by CIS Email to 71355,470. You can also mail credit card orders to PsL
at P.O.Box 35705, Houston, TX 77235-5705.
"THE ABOVE NUMBERS ARE FOR CREDIT CARD ORDERS ONLY.
"THE AUTHOR OF THIS PROGRAM CANNOT BE REACHED AT THESE
NUMBERS.

Other users: Complete the registration form (see regform.wri) and mail it along
with payment to:
Jeffrey Cooke
PO Box 106
Kemps Creeks
NSW 2171
Australia

If you wish to obtain complete source code via Email, include your Email
address, otherwise if you want to receive the source code on disk then add
US\$4.00.

If paying by check please make checks payable to Jeffrey Cooke.

Australian users should phone or send a fax for local registration details.

© Jeffrey Cooke, 1996.
compuserve: 100026,3107
internet: 100026.3107@compuserve.com
Fax: 61 47 749086