PxCaseStudio v.1.15

for ASP.NET and C#

Universal tool for modeling, recording data and generating functional database application running under Oracle, MS SQL, MySQL, Firebird, Interbase databases

Contents

1. Introduction

The PxCaseStudio is a universal tool for modeling and recording of data as well as for generation of the functional database application running under ASP.NET, with the C# programming language.

The PxCaseStudio allows you to design large-scale database applications for different databases in no time! Support includes database servers such as Oracle, MS SQL Server, MySQL, Firebird and Interbase server.

The PxCaseStudio generates SQL script and native code of the application in the C # language with individual forms, tables, grids, edit controls, menu and so on. The output code is largely built on the Px Framework components, where the Px Framework has been specifically developed to enable such generation of the application from the PxCaseStudio.

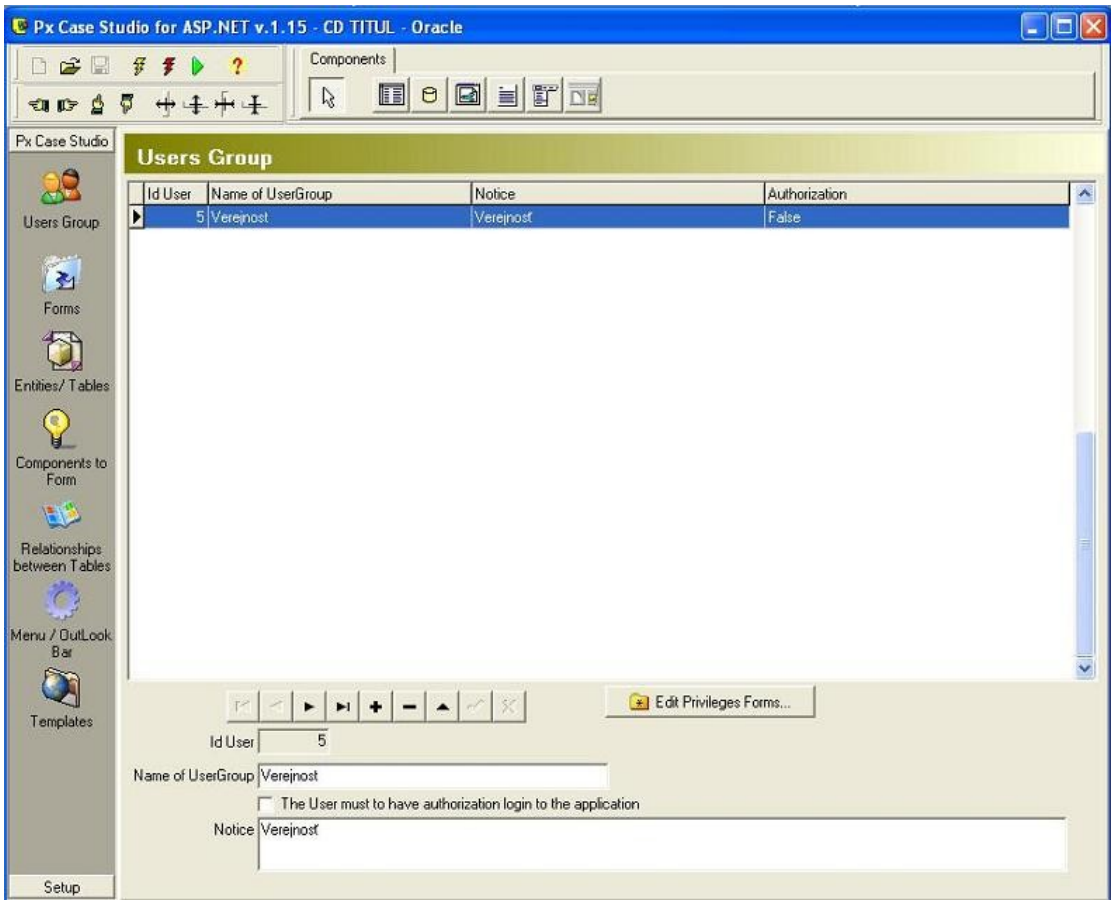
It deals here with the accurate modeling and the Px Framework components are the exact building bricks that allow such accurate modeling.

The Px Framework introduces the philosophy of stacking the configuration data, relationships and connections, different types of validations, etc. at the PxWebQuery database component via the AddParam ... method and its derivatives. This greatly speeds up the development of the application, improves its modifiability and clarity of the source code. If you want to use the PxCaseStudio effectively, we recommend you to study the philosophy and operating principles of the Px Framework components.

2. Specification of new user groups (Form „Users Group“)

The "Users Group" form is used to enter all user groups that should exist for the application you want to create. You can add a new user group by pushing the "plus" button.

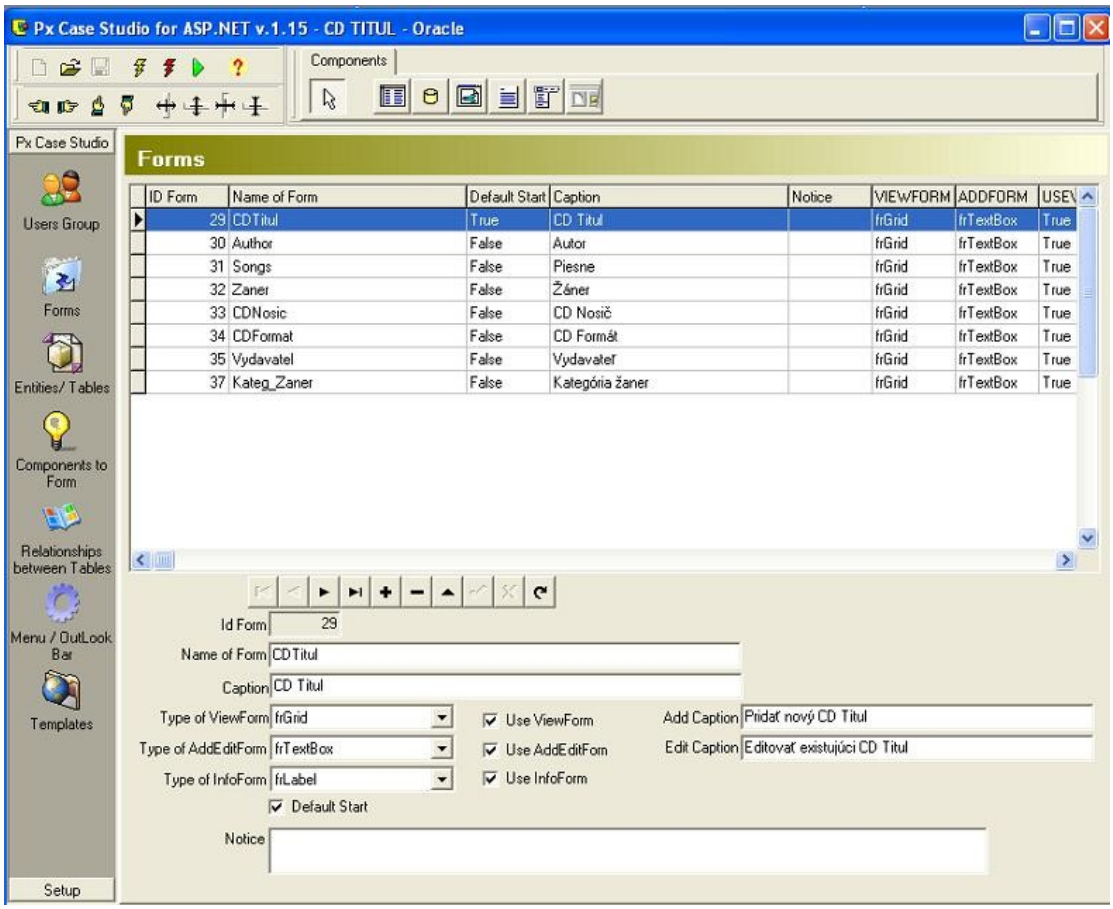
The "Edit Privileges Form" button opens the form in which you can assign privileges and access rights for individual forms, and thus assign permissions to each group. But this form should be opened only if all of the forms are set. The name of the group in the "Name of UsersGroup" heading shall be entered without diacritics and spaces.



3. Definition of forms (Form „Forms“)

This form serves for definition of all forms, their names, caption, as well as for checking whether you want to generate the View Form, AddEdit Form and Info Form. Thus, during the definition of one form, you can generate up to three different forms with different functions. The "View Form" is the form that contains the grid(PxSuperGrid) and is used to display the table data, their selection, filtering, sorting and so on. The "AddEdit Form" is used to enter and edit new data, and is formed by such editing elements as PxEdit, PxComboBox, PxJSDatePicker, etc..
The "Info Form" is used to display details of the selected row of the table, and formed by the PxLabel component for displaying data, and so on.

In the "AddEdit Form" you can enter individual titles that will be displayed at the top of the form. This can be done for the two states, i.e. table row insertion and editing. Furthermore, you can decide which of the three forms - the "Form View", the "AddEdit Form" or the "Info Form" you want to generate. If you want to generate all three forms, check all three forms.
Names of the forms in the "Name of Form" heading shall be entered without diacritics and spaces.



4. Definition of tables / entities (Form „Entities / Tables“)

In this form, you can define a table, view, etc.., and then you can define their structure. You can enter several types of entities, which list can be found in the "Type of Table". These are „Tables“, „ Code Lists“, „Query“, „Internal Query“ and „View“.

4.1. Addition or creation of a new "Table"

You can add a new table by means of the "plus" button. Fill in the name of the table without diacritics and spaces, and then specify the type of table "ttTable". Use the checked button to enter the data into the database. Then press the „Edit Structure Table“ button. This will open the „Add Structure Table“ form. Here you can specify individual table columns, their names without diacritics and spaces, text of the column as well as type and size of the column. Then you can check whether the column is the primary key or the "Not Null", or unique (IsUnique).

Each table shall contain a column with the primary key. For its proper functioning the Px Framework requires the table with a primary key.
When you create a primary key column, name it according to following rules: use the name of the table for which this column is created and put the word "ID" before it. For example, the primary key column for "CdTitul" table shall be called “IDCdTitul”.

Creation of a primary key column name:

ID+<Name of Table>

4.2. Addition or creation of a new "Code List"

The Code List is created exactly like the table, but for the type of table you shall select the "ttCodeList" value. Table columns are specified just like in case of the table and do not forget the primary key column.

4.3. Addition or creation of a new "Query"

If the table is already created in the database, you can use this table via the type of entity, which is "ttQuery".

The query is a clean SQL command in the form of "select * from table_name", which you write in the "SQL Script" heading.
So when filling the fields you shall enter only the caption, i.e. the field Caption.

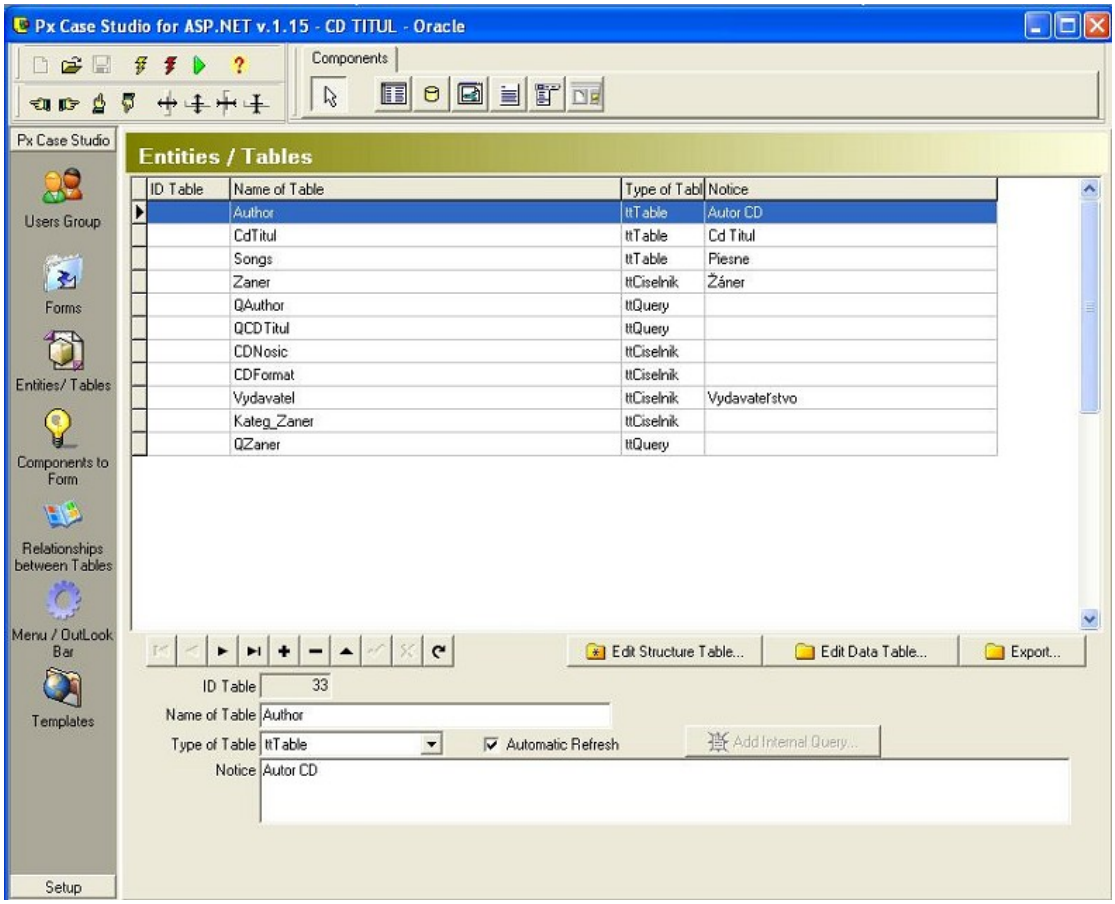
4.4. Addition or creation of a new "Internal Query"

If you have entered the table of the "ttTable and" ttCodeList " type in the PxCaseStudio, you can use it to specify the " ttInternalQuery "entity. For the "Internal Query" enter a clean SQL select, already over the existing tables that are set in the PxCaseStudio. To do this, use the "Add Internal Query" button. Pressing this button will open a form where you can select the table and check-mark individual columns, from which the specific SQL command shall be created.

So when filling the fields you shall enter only the caption, i.e. the field Caption.

4.5. Addition or creation of a new "View"

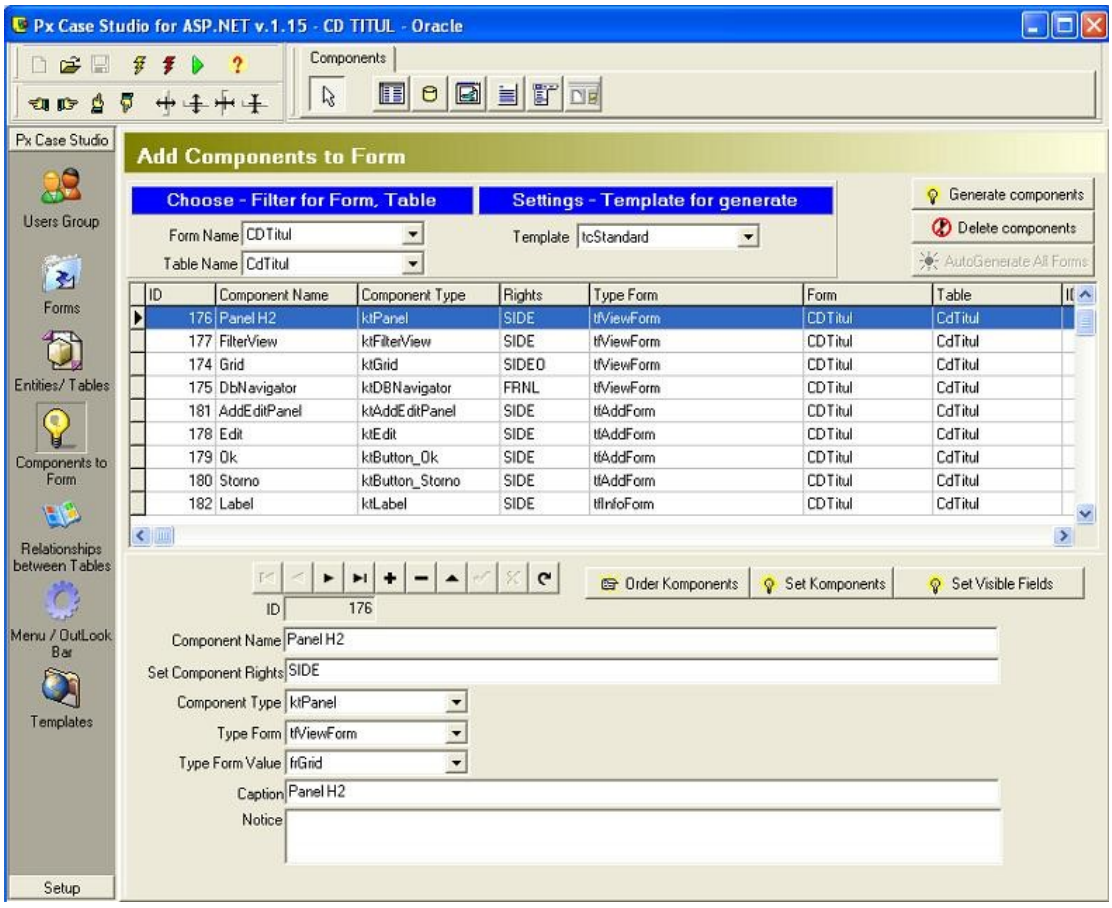
If you create an entity of the "ttTable" and "ttCodeList" type these entities shall be generated as the database table during the final SQL Script generation.
In case of the "ttView" entity the SQL command that you have entered into the "SQL Script" heading shall be generated or entered.
This type of the entity type is suitable for View definition. Table columns are left blank.



5. Definition of form and table interconnection (Form „Component to Form“)

In this form takes place the pre-generation, individual Tables, Code Lists, Query, etc. are connected to the individual forms.
First, you choose the form and the table that shall be output to the given form and select the type of its template.
Then push the "Generate components" button. Repeat this pre-generation individually for each form.

After pushing the "Generate components" button, individual elements and components for the given form shall be generated according to the template. These components connect the form with individual tables in the database. If you want to generate a form with a Code List, use a template called "tcCodeList”. In other cases use the default template "tcStandard".



By means of the "Order components" button choose the components display order in the direction from top to bottom in the form, where they shall be displayed.

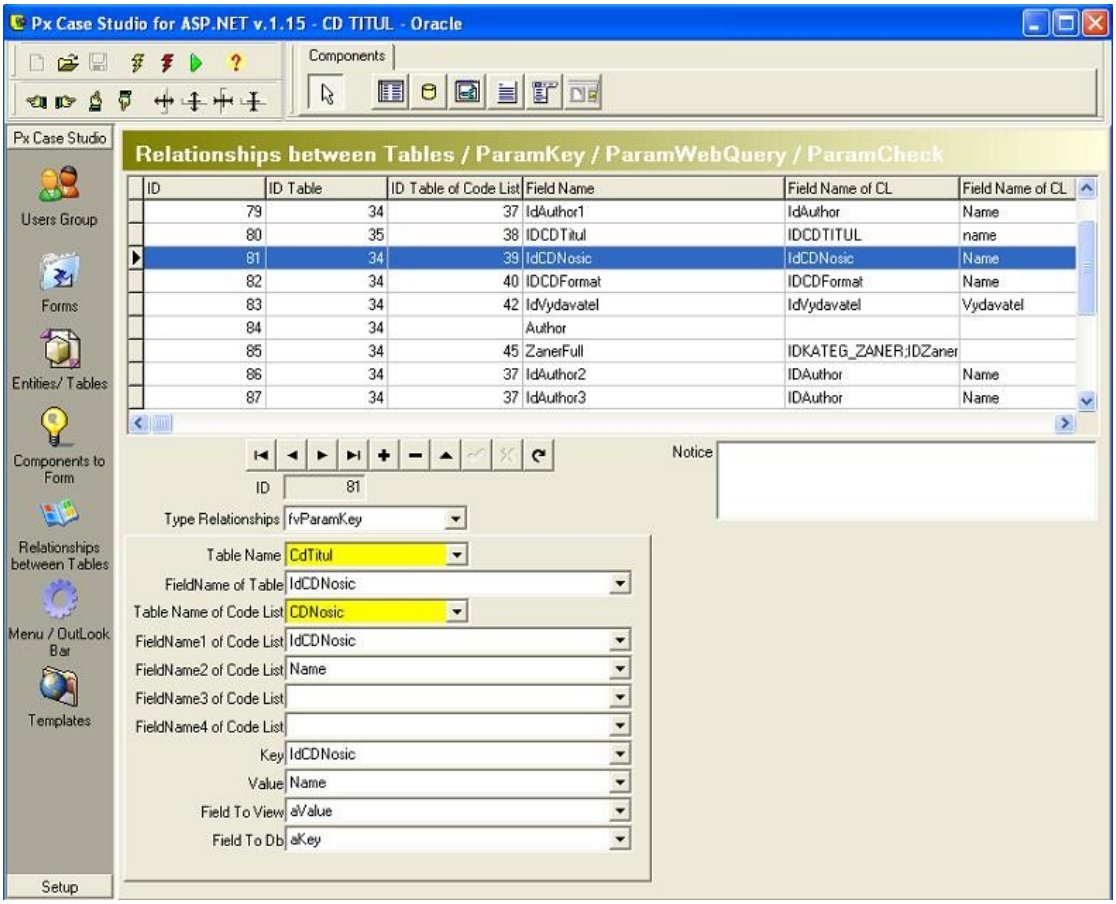
Use the "Set Visible Fields" button to define which items, columns shall be displayed in the individual forms namely in the "View Form", "AddEdit Form" and "Info Form". In the "View Form" the number of columns displayed is reduced in case they are too many and only the most important ones are displayed. The "AddEdit Form" and "Info Form" display mostly all columns that we want to edit or display.

6. Definition of links between tables (Form „Relationships between Tables“)

This form is used to create links between individual entities, tables. Here, you can actually specify the relationship between tables .
During generation these individual relationships are reflected in the generated application.

The simple relationship between the Table and the Code List is represented by the "AddParamKey" and "AddParamWebQuery" relationship. The More complex relationship that allows you to specify the category and sub-category is the "AddParamFlyComboBox".

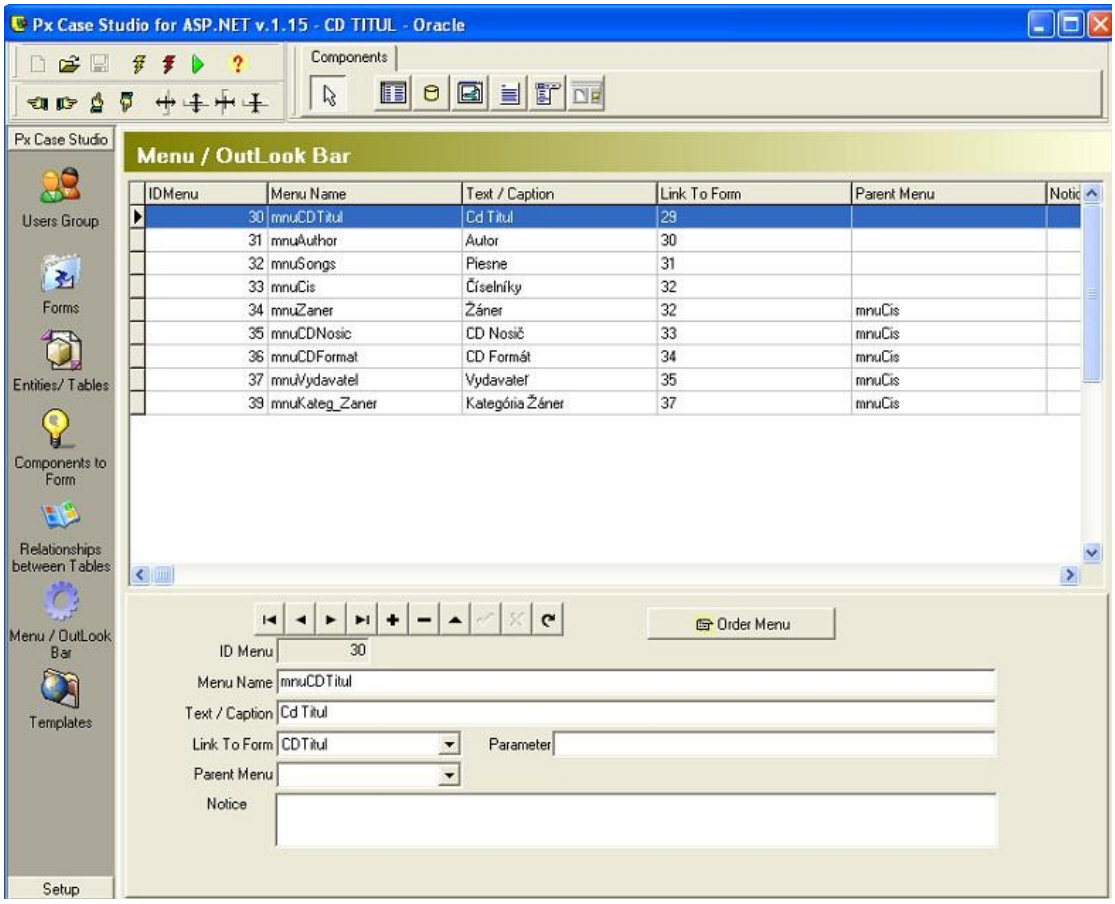
For these individual relationships we recommend you to study individual parameters of the PxWebQuery component (AddParam. ..) and its derivatives. The PxWebQuery component is the Px Framework component.



7. Creation of an application menu (Form „Menu / Outlook Bar“)

This form is used to create a menu and its linking to individual forms.

In the "Menu Name" line enter the menu name without diacritics and spaces.
In the "Text / Caption" line enter the real menu name that shall be displayed in the application.
In the "Link to Form" line select a form to which the menu shall be linked.
Clicking on the menu shall open a form that is specified in “Link to Form " line.
If you want to create a sub-menu, do it via the “Parent Menu" line, where you shall enter the name of the root menu. If you want to sort the menu items in the specific order, use the "Menu Order" button.



8. Final generation of the application

When all data is entered and mutually interconnected, you can start final generation, compilation of the application. But prior to this, enter the application settings, and set the Connection String, title and name of the application as well as the type of the database for which the application shall be generated. You can choose from the following databases: Oracle, MS SQL, MySQL, Firebird and Interbase.
The following example contains the description of the keys, which are used for the final application content generation:

 - is used to generate the final SQL Script

 - is used to generate the final application code in ASP.NET and C#

 - is used to generate SQL Script, and the final application code in ASP.NET

The final generated code and Sql script are saved in the directory which was specified during the creation of the project.
The output directory can be found during the opening of the project by moving the table cursor to the left, the output directory is listed in the "Output Dir" column. In the settings you can choose for which type of the server you want to generate the application. If you change the server type, you have to completely delete the "Output Dir" directory and run the project compilation again.

When the whole application is generated, it should be located in the "Output Dir". This directory should also contain the "SQLScript.sql" file.
Take this file and create the database for generated application.

Then run the MS Visual Studio and open the generated project via the "**Open website ... "** command.

The "CD Title" project is loaded in the PxCaseStudio installation.
This project is the fully functional example of how to create a fully functional database application by means of the PxCaseStudio.
If you want to generate a final project, you should, prior to its generation, choose in the settings for which database type this project shall be generated.

We wish you success in working with the PxCaseStudio application.

Relationships between individual tables

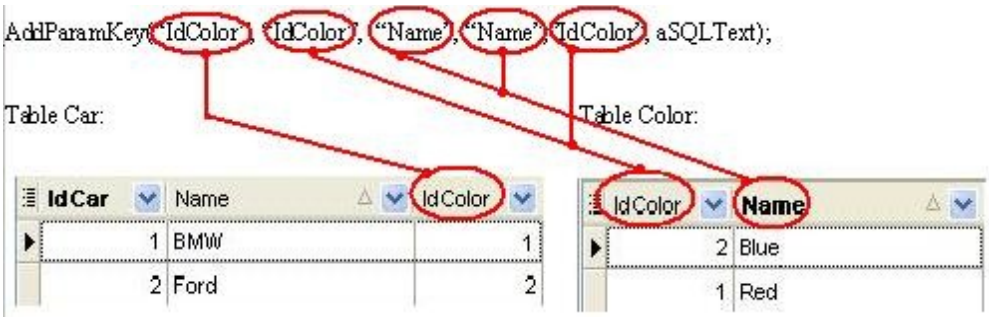
The AddParamKey() parameter of the PxWebQuery component:

Definition fields:

AddParamKey(aFieldName, aFieldNameKey, aFieldNameValue, aFieldToView,aFieldToDB, aSQLText);

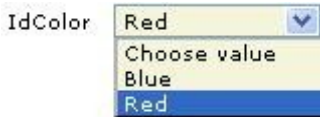
aSQLText = “select IdColor, Name from Color”;

PxWebQuery.AddParamKey(“IdColor”, “IdColor”, “Name”, “Name”,”IdColor”, aSQLText);

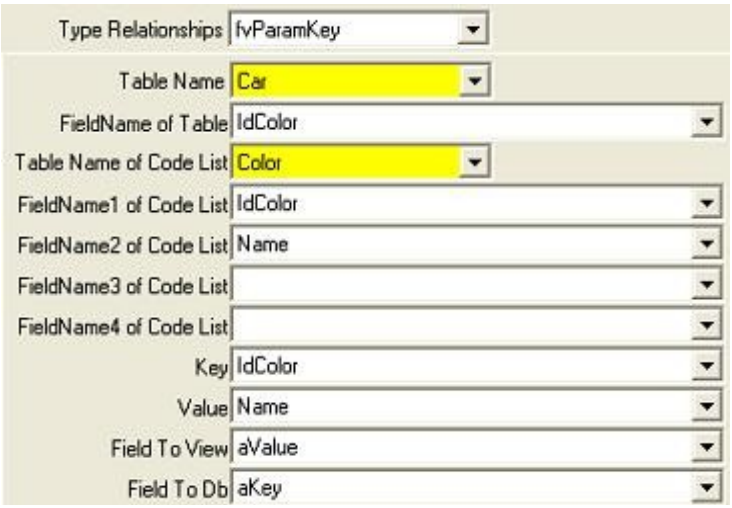


Output in the PxSuperGrid component:

Output in the PxComboBox component:



Definition of the AddParamKey parameter in the PxCaseStudio:



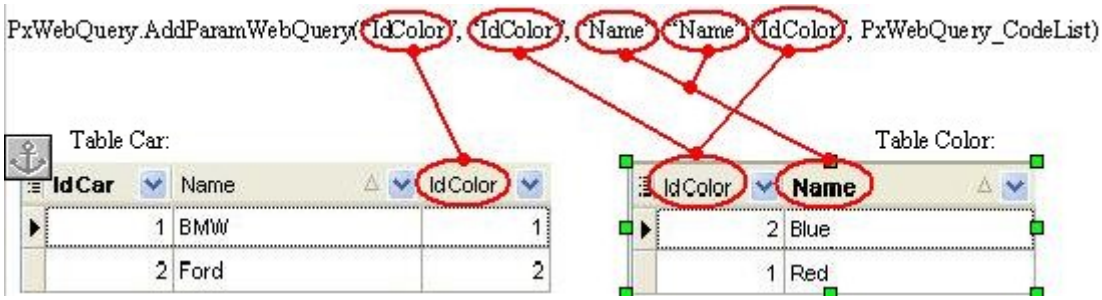
The AddParamWebQuery() parameter of the PxWebQuery component:

Definition fields:

AddParamWebQuery(aFieldName, aFieldNameKey, aFieldNameValue, aFieldToView,aFieldToDB, PxWebQuery_CodeList);

PxWebQuery_CodeList.SQLSelect = “select IdColor, Name from Color”;

PxWebQuery.AddParamWebQuery(“IdColor”, “IdColor”, “Name”, “Name”,”IdColor”, PxWebQuery_CodeList);



Output in the PxSuperGrid component:

Output in the PxComboBox component:

Definition of the AddParamKey parameter in the PxCaseStudio: