Hide Data in Software

Last update: March 15, 2015

Contents

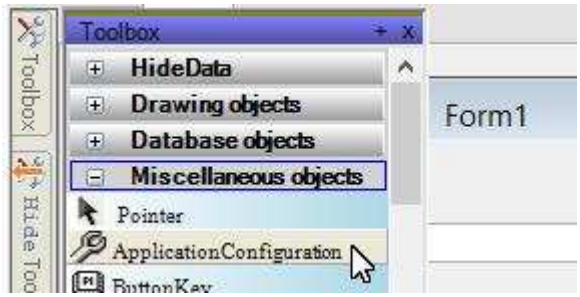
Introduction

When you distribute your software to users, your software may contain some sensitive information you do not want the users know. For example, suppose your software uses an account and password to access a database for some functionality. The account name and password are contained in the software, say, in an EXE file. Someone may disassemble the EXE file and find out the account name and password. You certainly do not want that to happen.

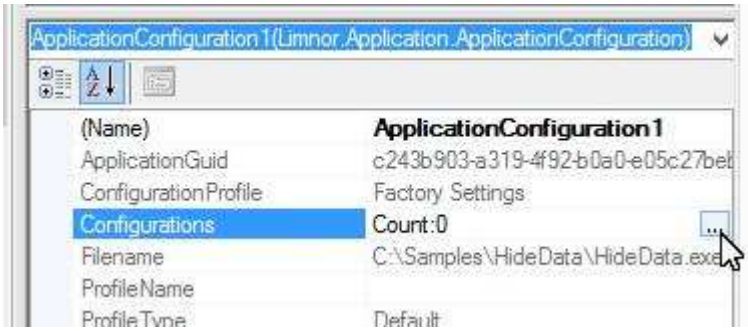
The Application Configuration component provides data encryption support. You may use it to hide sensitive information in your software so that only your software may use the information. This document shows a sample of doing it. But remember that any data stored in a computer is at various degrees of risks of being decrypted, and it is your responsibility to evaluate degrees of risks of applying different techniques and accept consequences. Please read Limnor Studio software license for details.

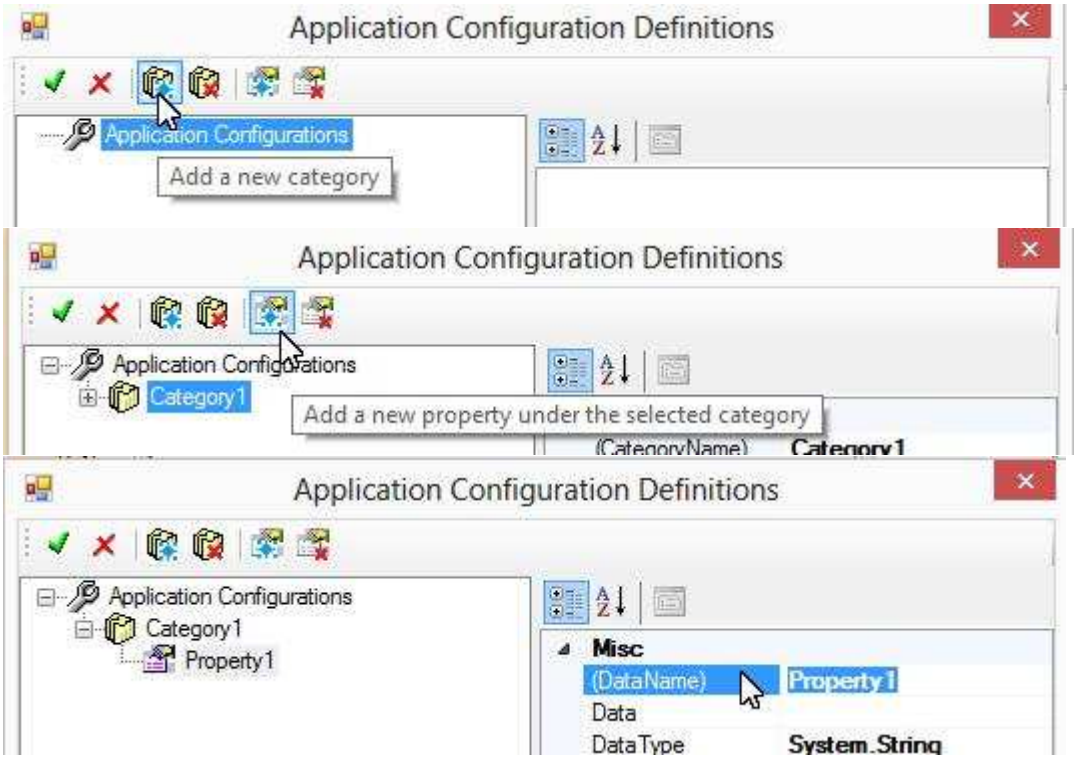
Encrypt Data

Drop an Application Configuration component to a form where some sensitive information is used.

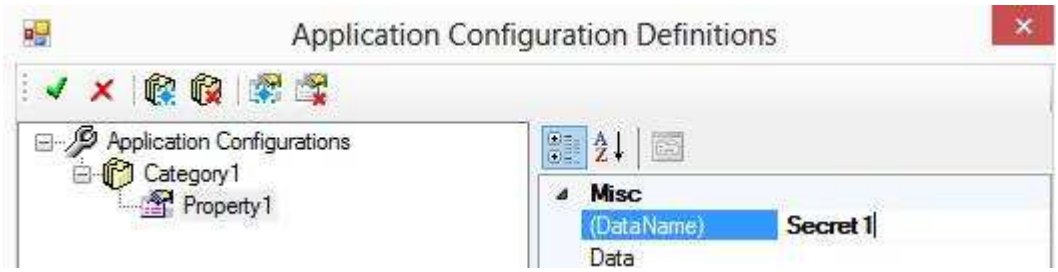


Create one configuration value for each piece of sensitive information. For more information on creating configuration values, see http://www.limnor.com/support/UseAppConfig.pdf.

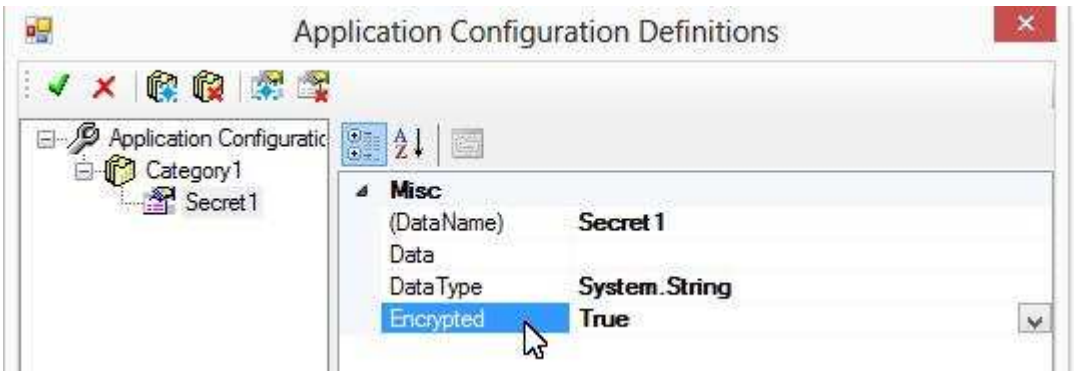




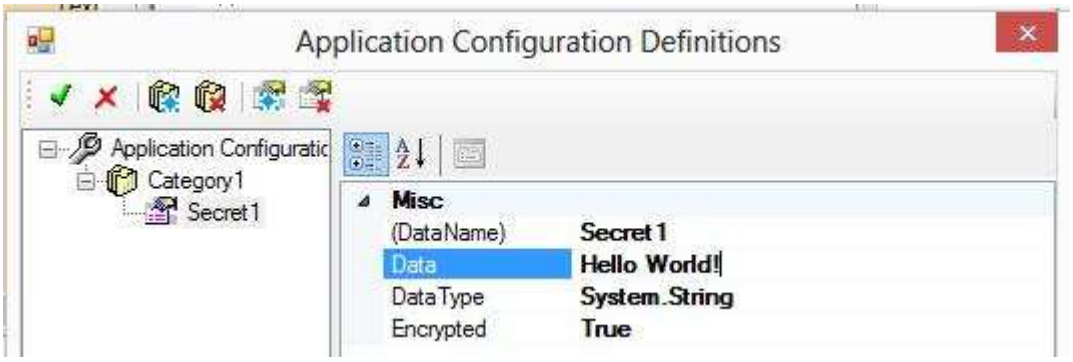
You may change the name of the value, for example rename the value to “Secret1”:



Set its Encrypted property to True:



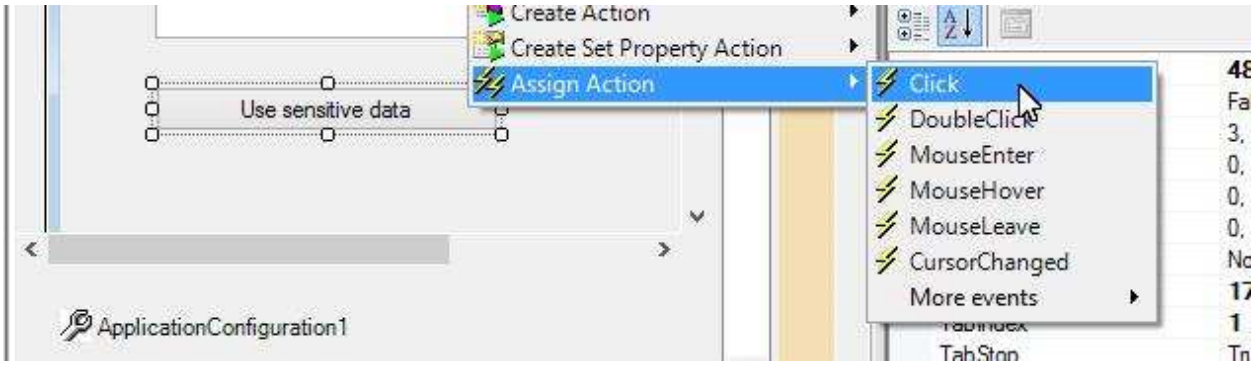
Enter our sensitive information. In this sample, it is “Hello World!”:

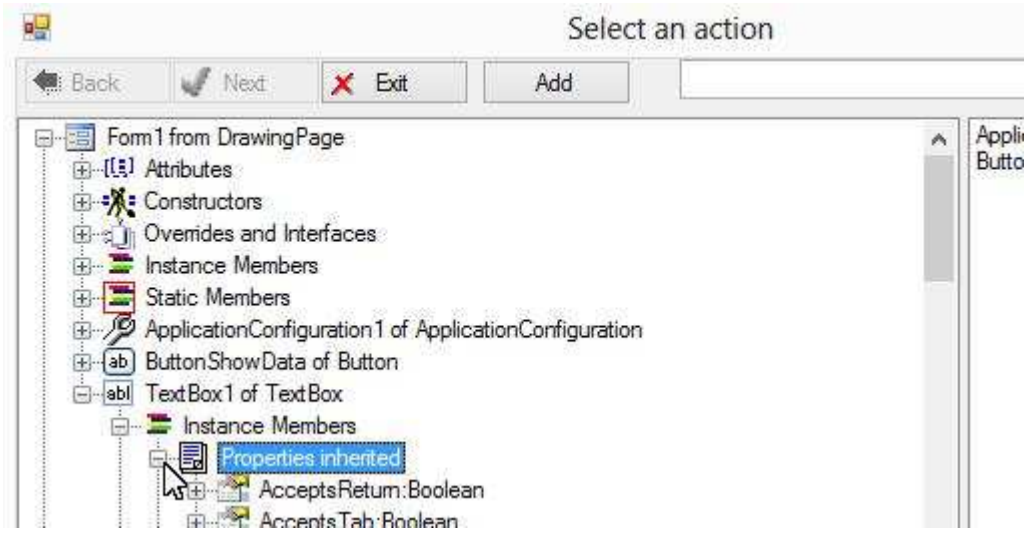


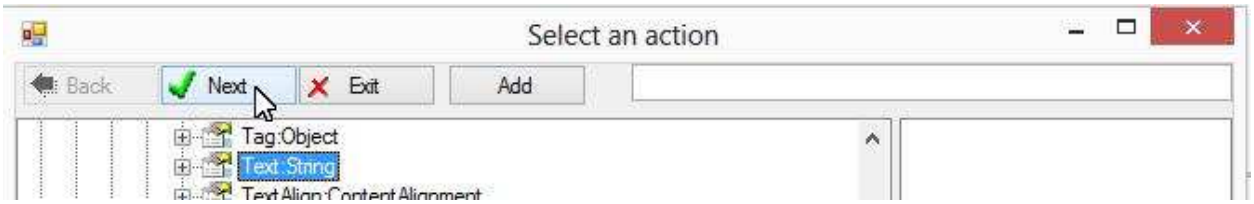
Now, we hide one piece of sensitive information. We may create more vales if needed.

Use Hidden Data

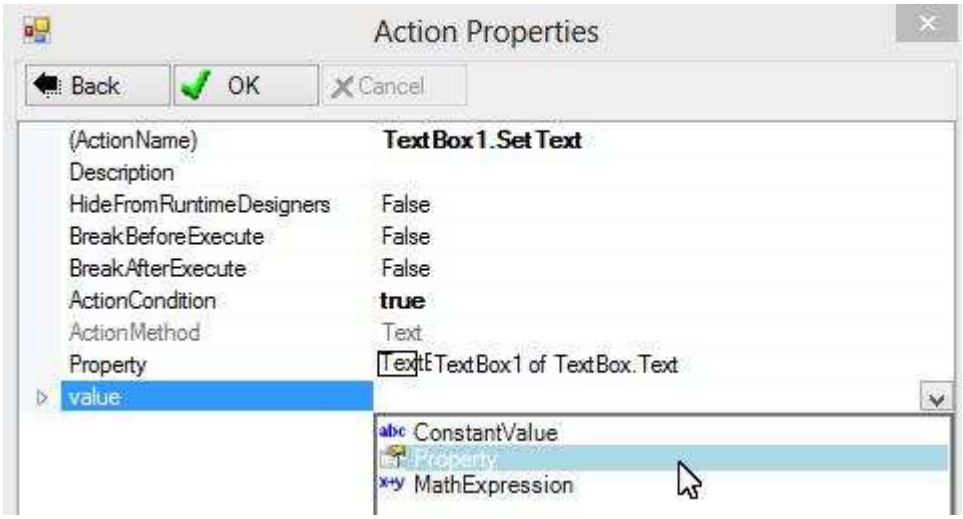
There is nothing special about using hidden data. They are used the exactly same way as unencrypted data. In this sample, we show the value of “Secret1” in a text box. For most sensitive information you probably do not want to do that.



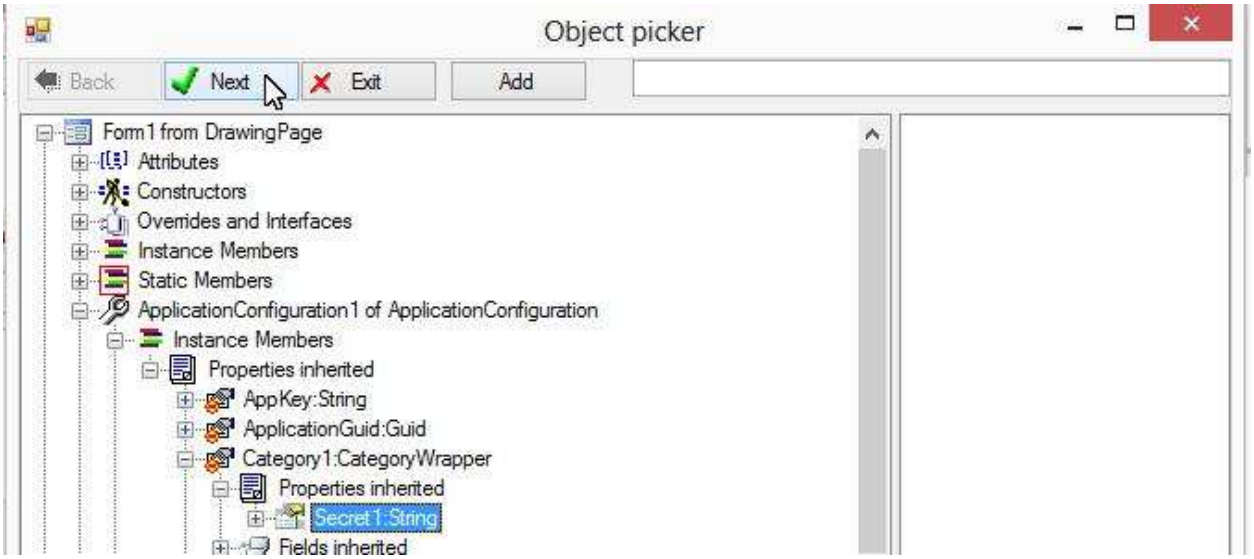




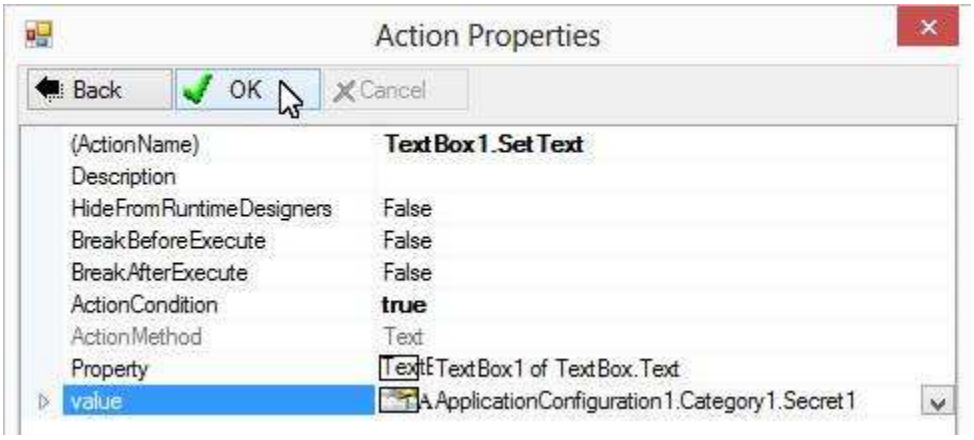
Choose “Property” to access a configuration value:



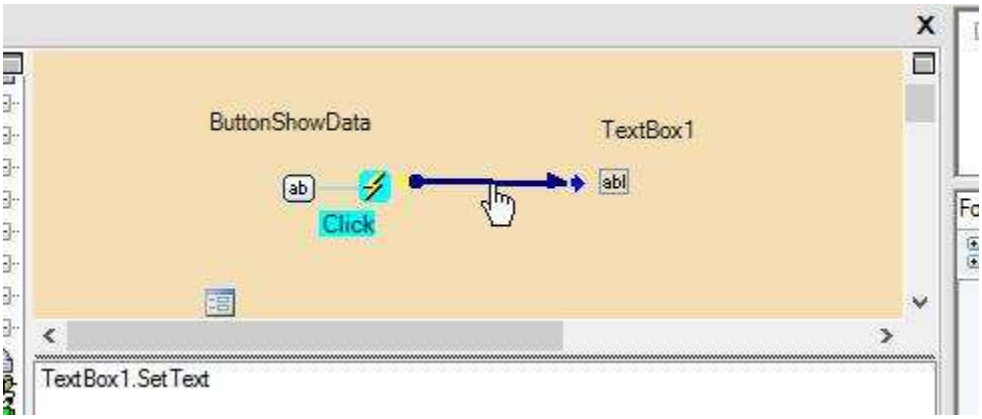
Select the configuration value to be used:



Click OK:

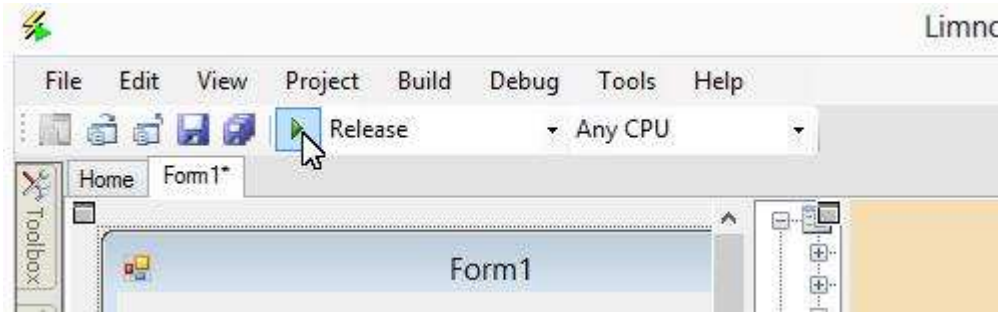


The action is created and assigned to the button:

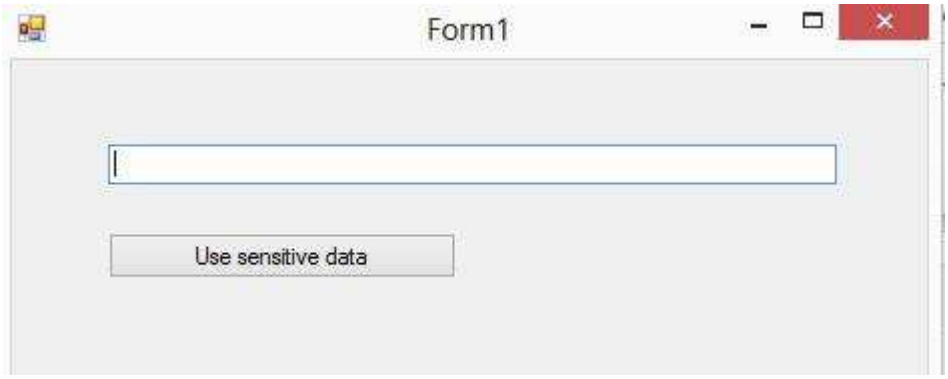


Test

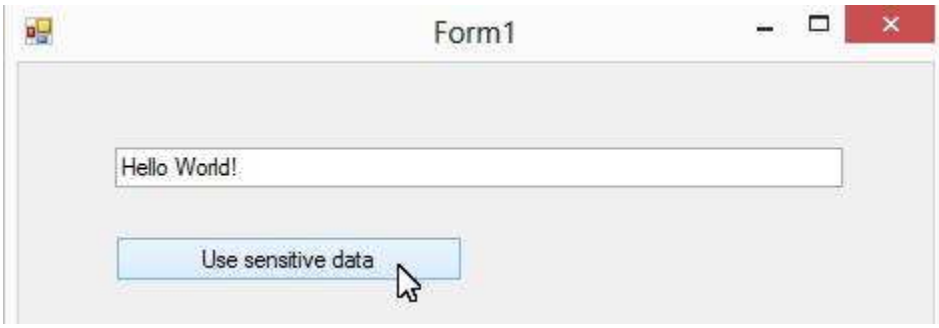
We have done our programming. Run the project to test it:



The form appears:

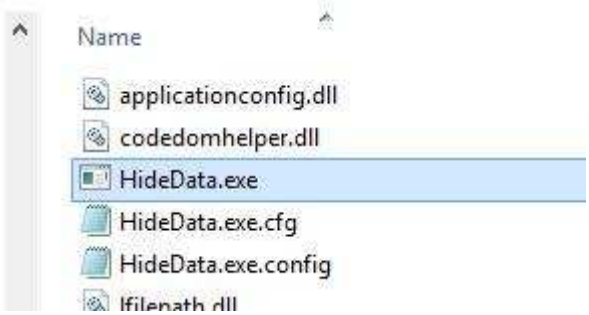


Click the button. The secret information appears in the text box:

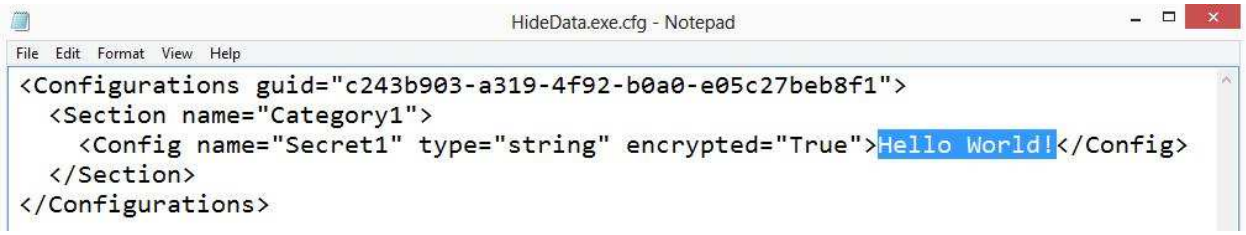


Data Protection

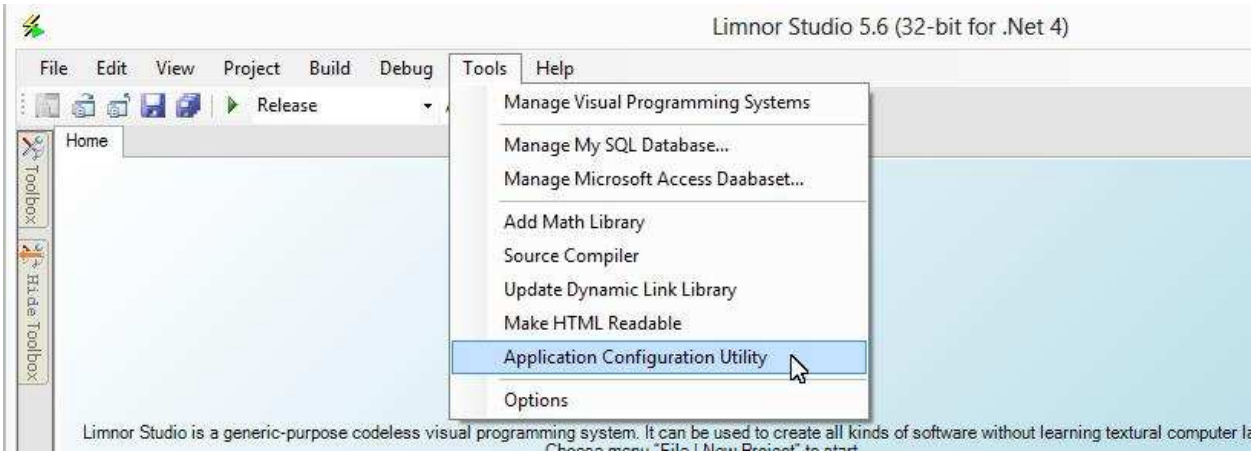
The sample project compiles to an EXE file:



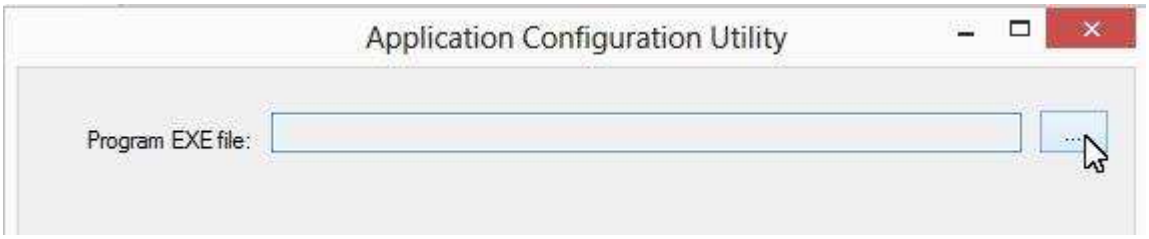
As our test shows, clicking a button, “Hello World!” appears. But the string “Hello World!” is not inside HideData.exe in any format. Therefore, disassembling HideData.exe will not reveal this string. Actually this string is in the file HideData.exe.cfg:



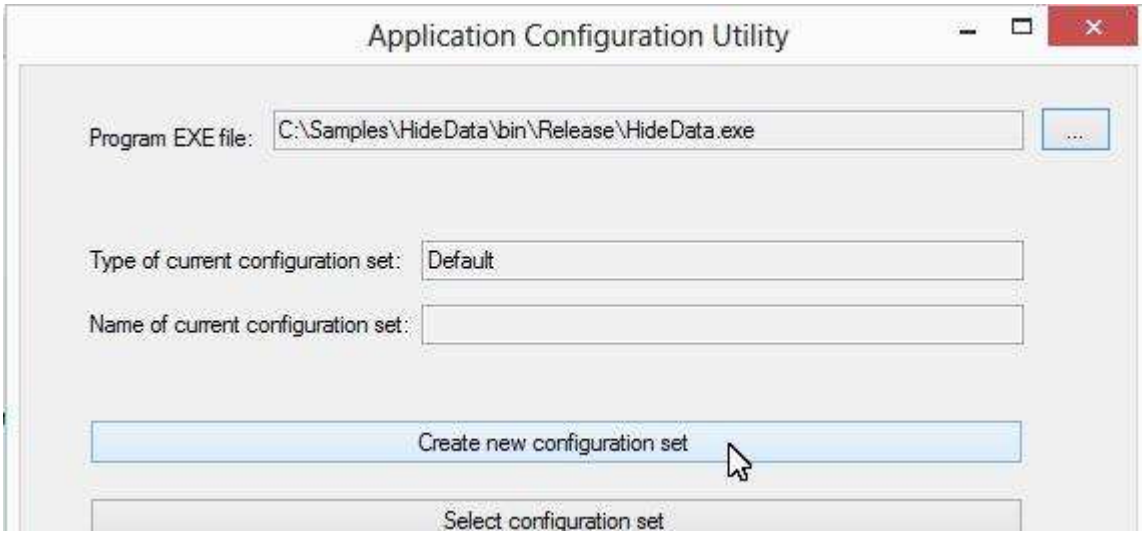
"Wait a second", you may say, "The words are not encrypted in HideData.exe.cfg". You are right, they are not. So, never give that file to the users. We need to protect the file. We may do it by "Application Configuration Utility":



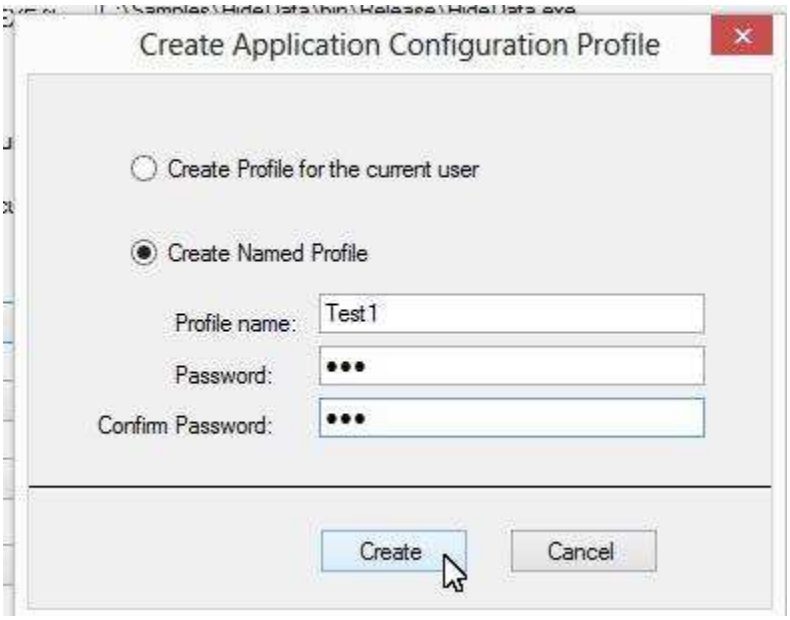
Select the EXE file:



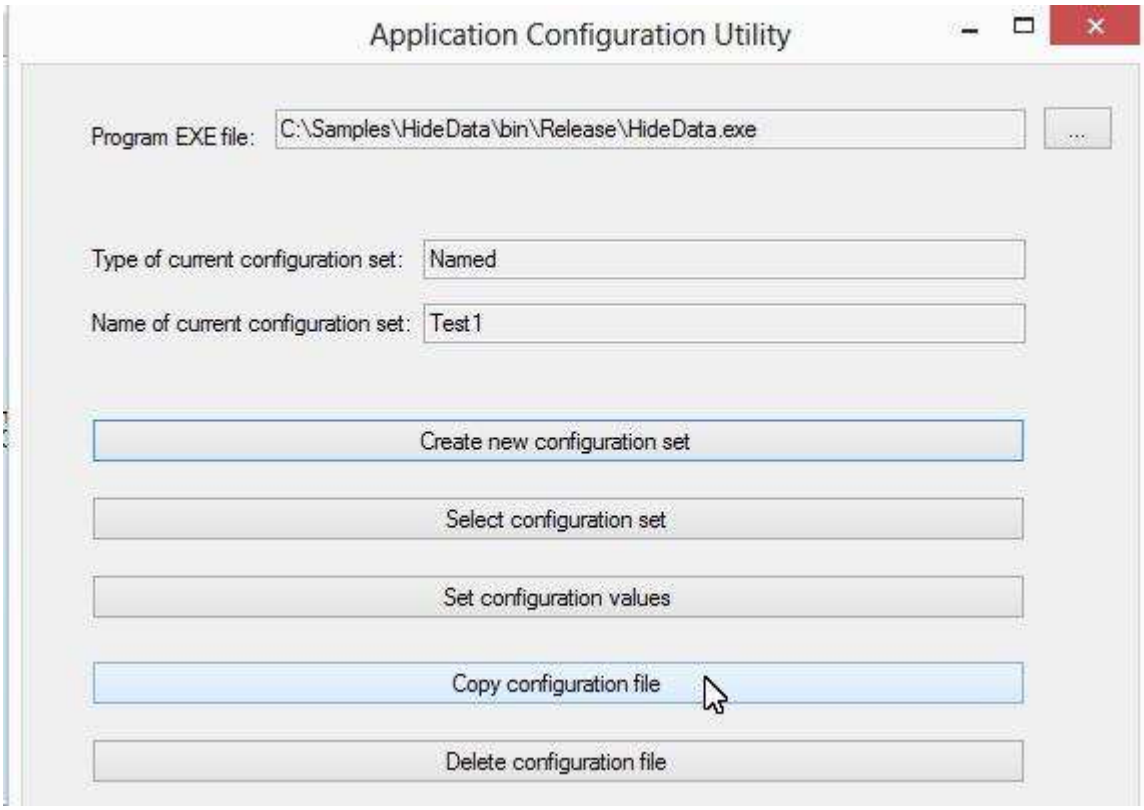
Click "Create new configuration set":

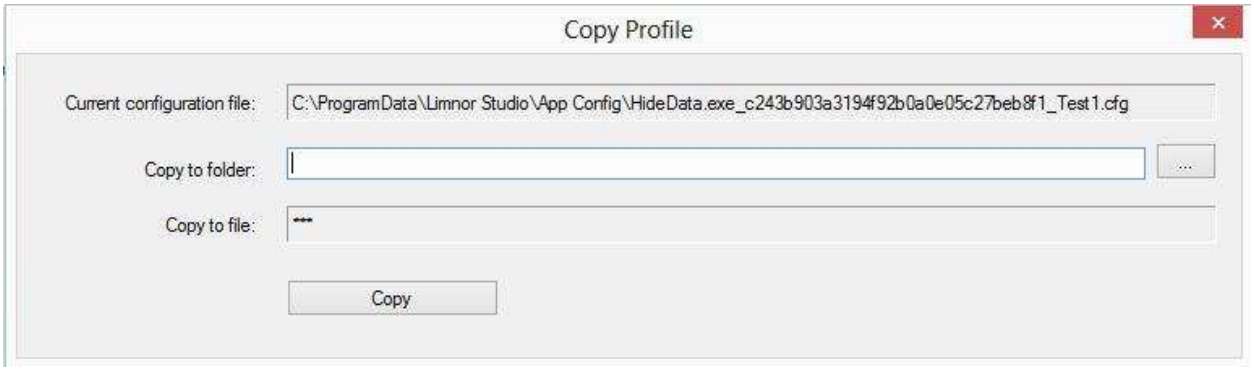


Choose “Create Named Profile”; give a name and a password:

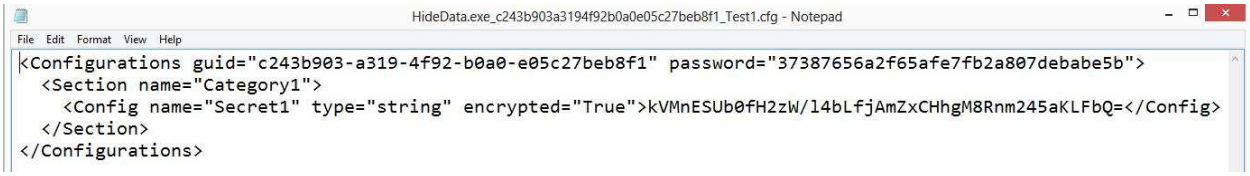


A new configuration file is generated. Click “Copy configuration file”, you can see where the file is and you may copy it to the folder to be distributed to your users:

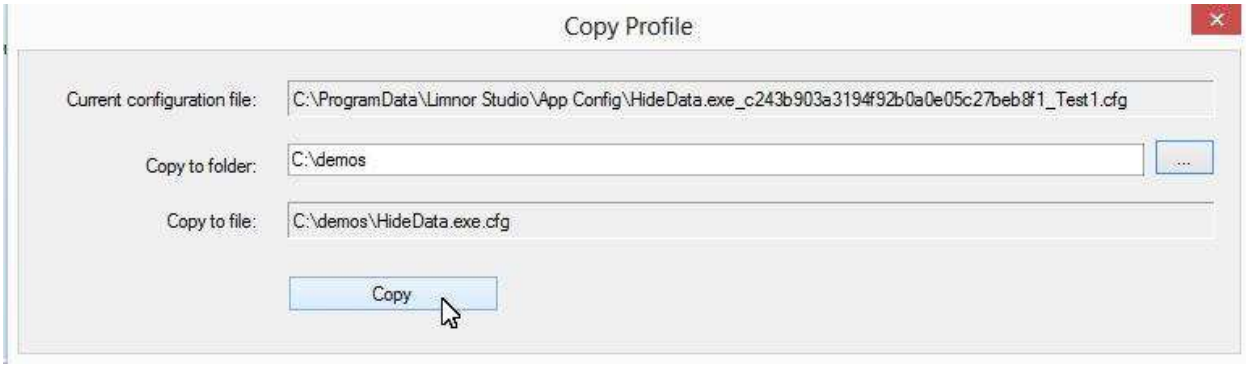




If you use a NotePad to open the new configuration file then you can see that the values with Encrypted being True are encrypted in the file:



You may specify a folder and copy the configuration file for distributing it to the users:

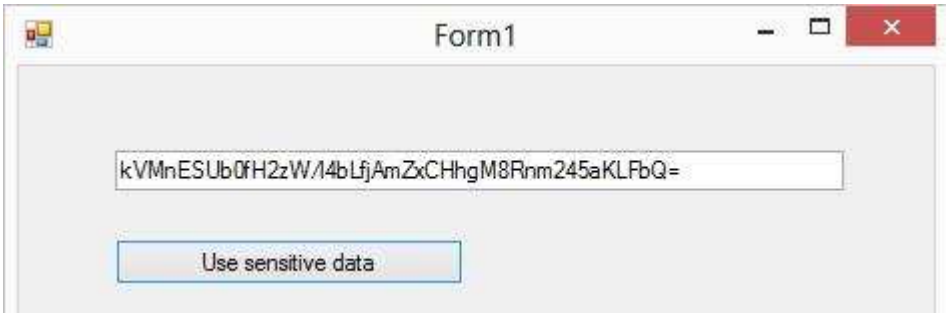


In a user's computer the CFG file should be in the same folder as the EXE file. For the above sample, the file HideData.exe.cfg should be in the same folder as the file HideData.exe.

You may copy the encrypted configuration file to the folder where the EXE file is compiled, for example, bin\Release under the project folder. But note that every time you make a new compilation, the configuration file will be overwritten with an unencrypted file. So, you need to do the above copying just before you are going to distribute the files to users.

Request Logon

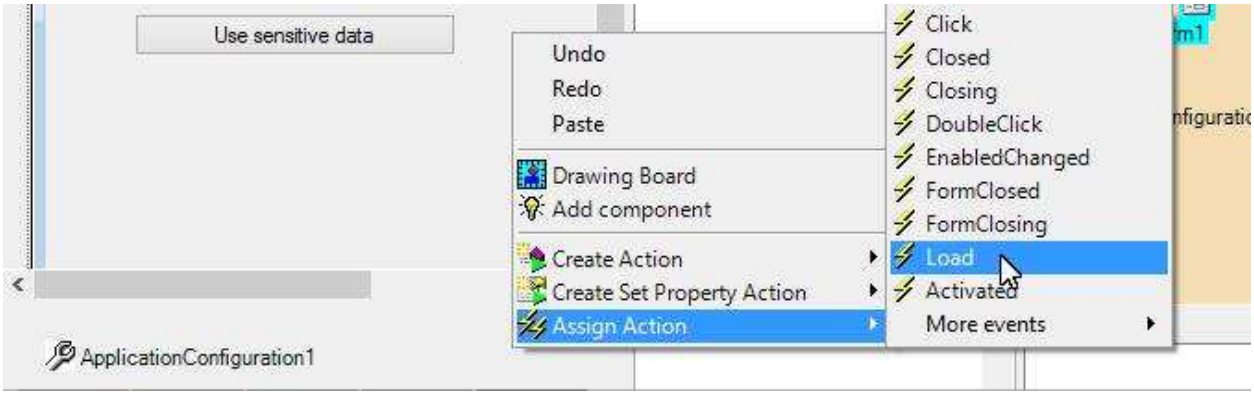
Let's copy the encrypted HideData.exe.cfg to the folder where HideData.exe is and run HideData.exe. Click the button to access the value of Secret1, and we get encrypted data:

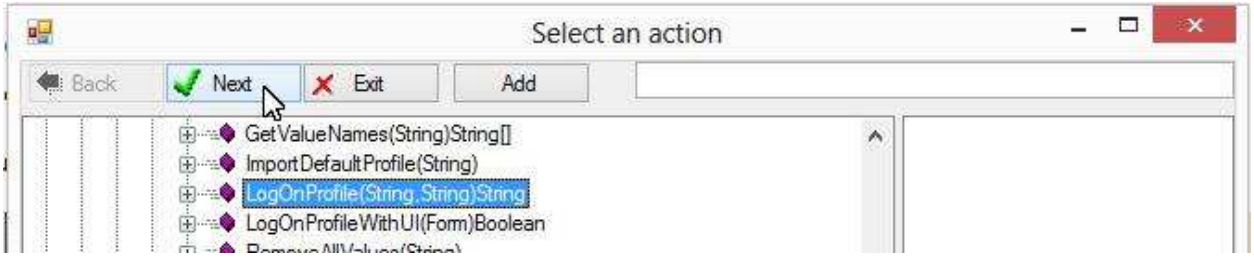


That is not what we want. We need to tell our software to decrypt it by logon with correct name and password.

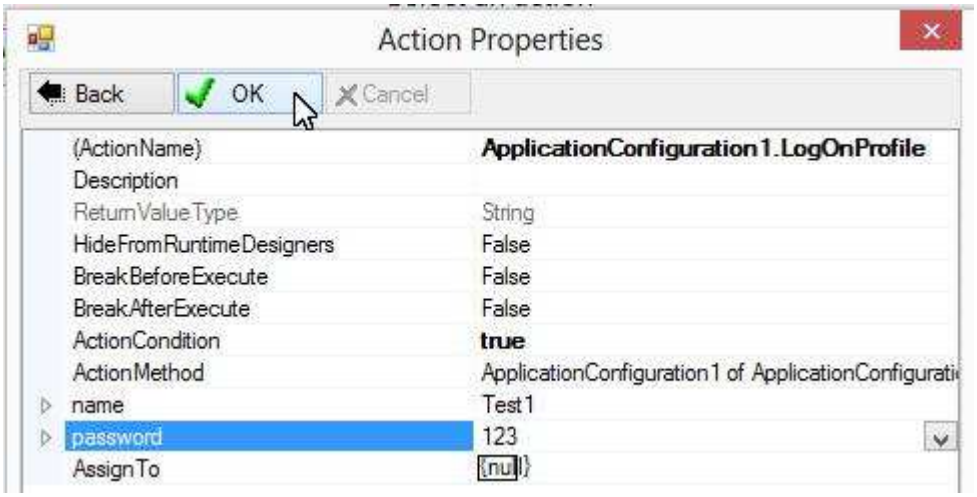
Automatic Logon

At the Load event of the form, we may execute a LogOnProfile action to logon to the encrypted configuration.

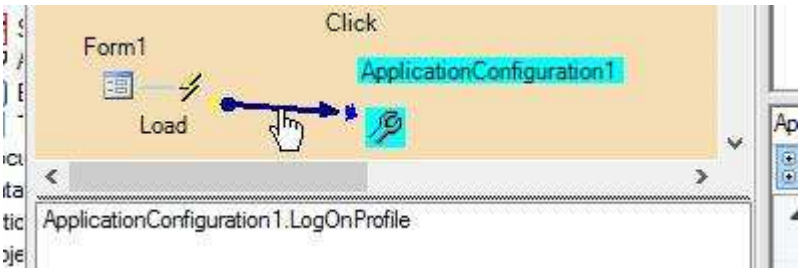




Specify configuration name and password for the action:



The action is created and assigned to the Load event:



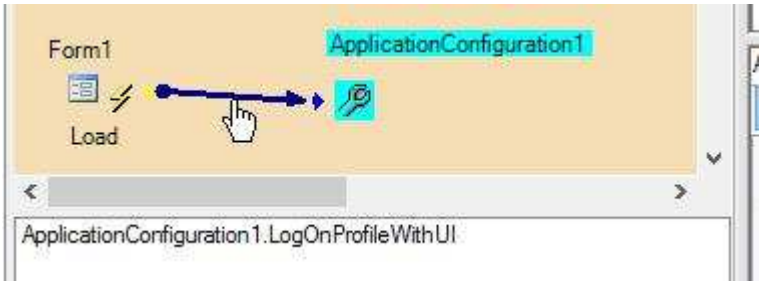
Compile the project to get new HideData.exe. Use encrypted HideData.exe.cfg to replace the one in the same folder of HideData.exe. Run HideData.exe and click the button. You can see that the value is decrypted:

In this arrangement, everyone can run your software.

Manual Logon

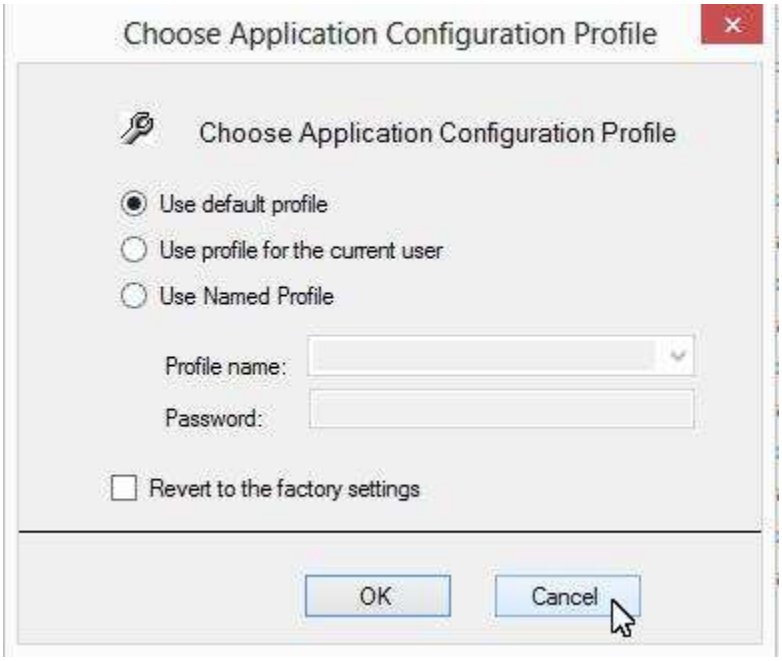
You may let the user to enter password and thus only those users knowing the password may run your application properly.

Remove LogOnProfile action from the Load event and assign a LogOnProfileWithUI action instead:

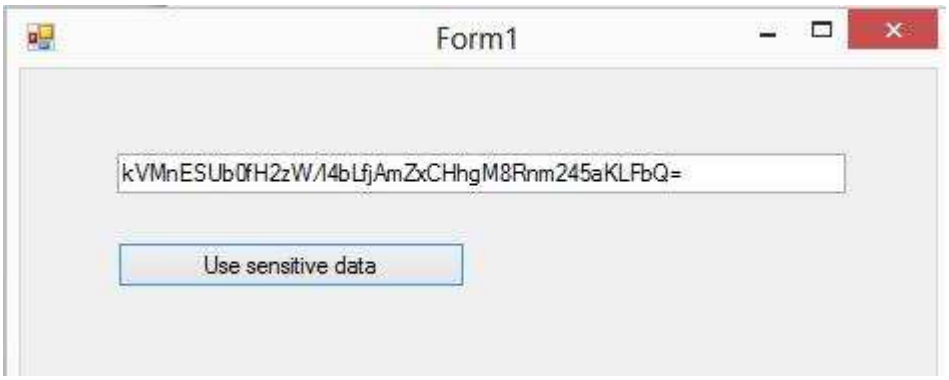


Compile the project and copy encrypted CFG file to the folder where the EXE file is.

Run the EXE file, a logon dialogue box appears:

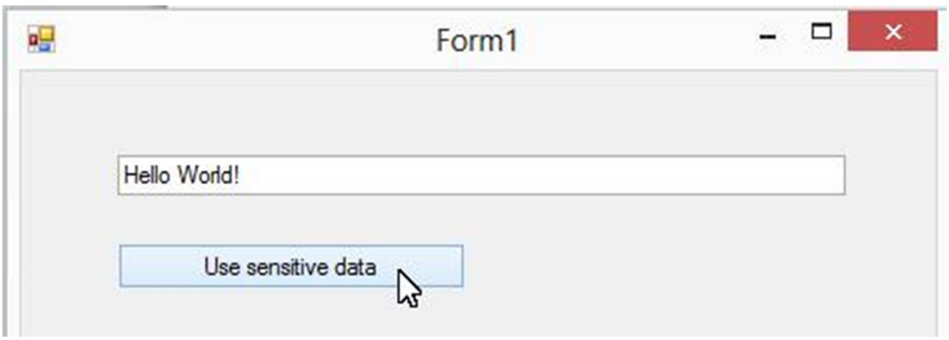


If we cancel the logon then the information will stay in its encrypted format:



Run the EXE again. This time use correct logon password:

Now the information will be decrypted when used:

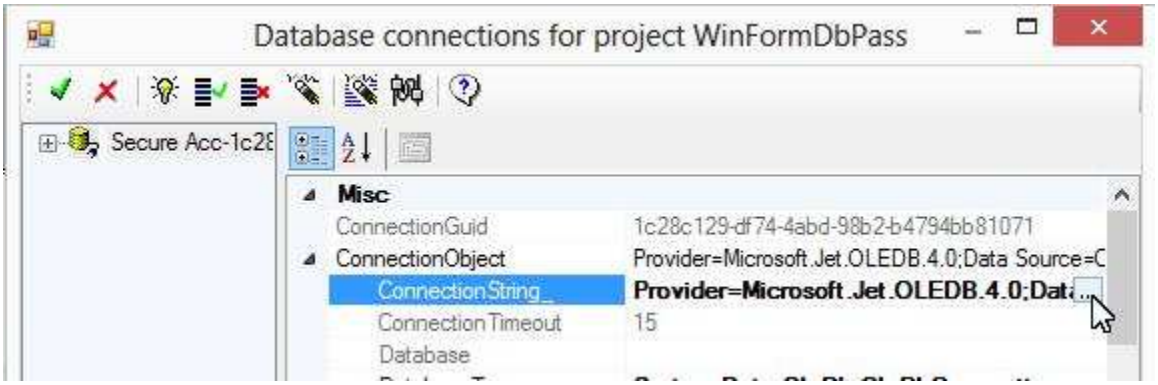


Use Hidden Password to Connect Database

In this section, we use the techniques presented previously to a practical usage of hiding database password.

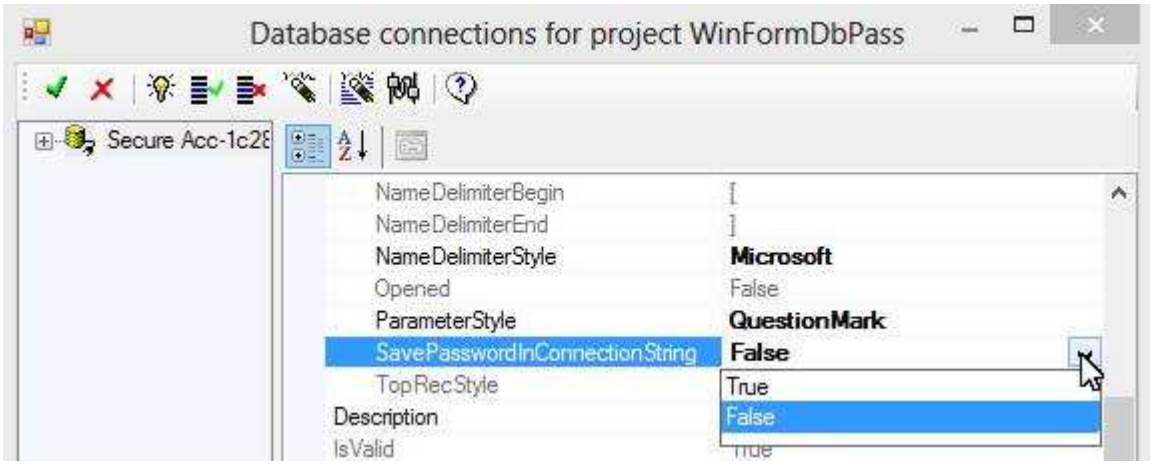
Use password at design time

Suppose we use a Microsoft Access database with database password "789". At design time, a developer needs to provide the password to the database connection settings:


Database connections for project WinFormDbPass
Secure Acc-1c28
Misc
ConnectionGuid
1c28c129-df74-4abd-98b2-b4794bb81071
ConnectionObject
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C
ConnectionString_
Provider=Microsoft.Jet.OLEDB.4.0;Dat
ConnectionTimeout
15
Database


Access database
Choose Access database
Database file:
C:\demos\dbSecure.mdb
Read only
Database password:
User name:
Password:
Test
OK
Cancel
Passwords are for testing the connection only. They are not saved

We set "SavePasswordInConnectionString" to False so that the password will not be compiled into the program.

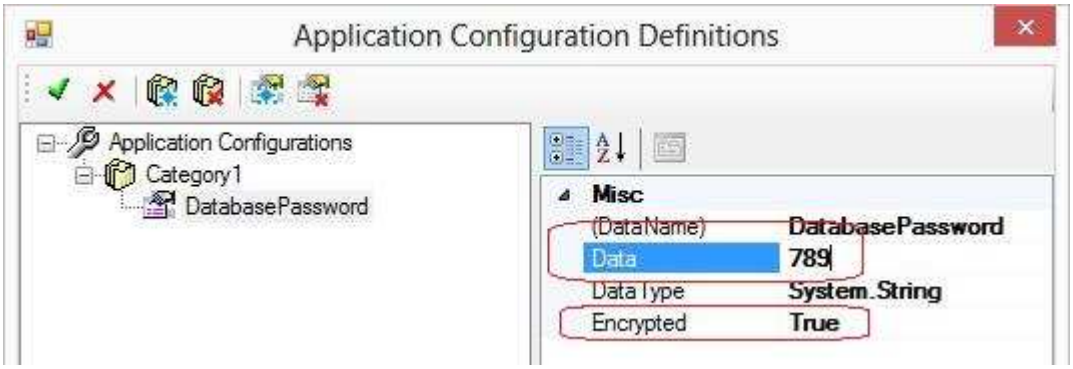


Use password at runtime

Since the password is not compiled into the program, we need to provide the password at runtime.

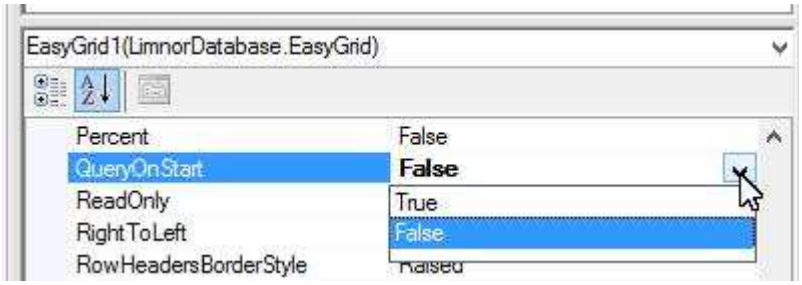
Use Application Configuration to hold password

We use the techniques described previously to save the password in a configuration file and encrypt it:



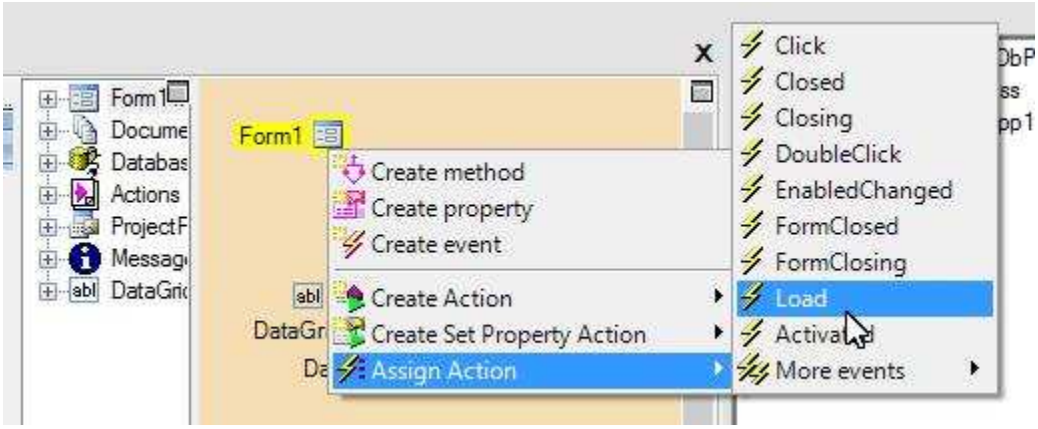
Disable automatic query

Set QueryOnStart of an EasyGrid or EasyDataSet to False so that it will not execute a query without setting database password first:

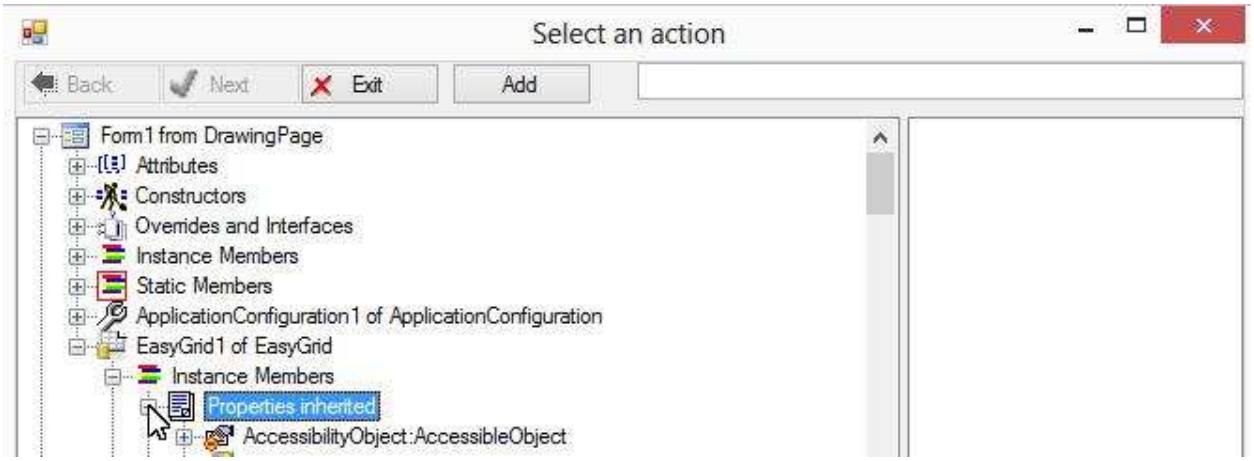


Apply database password

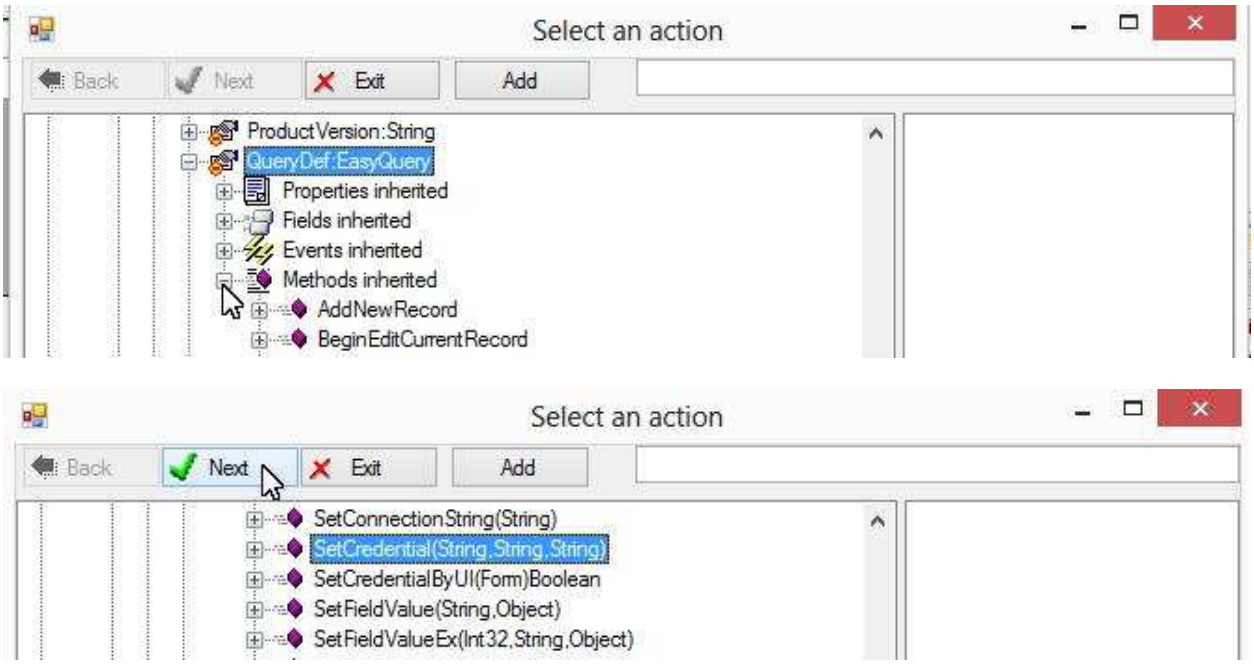
Suppose we want to query the EasyGrid at the event of form loaded. We first apply the database password at form load:



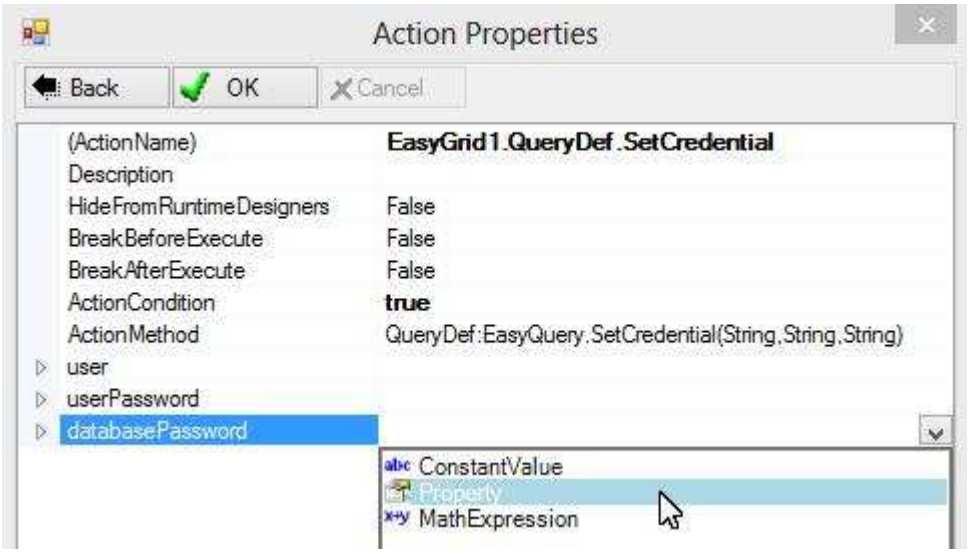
Find the QueryDef property of the EasyGrid:

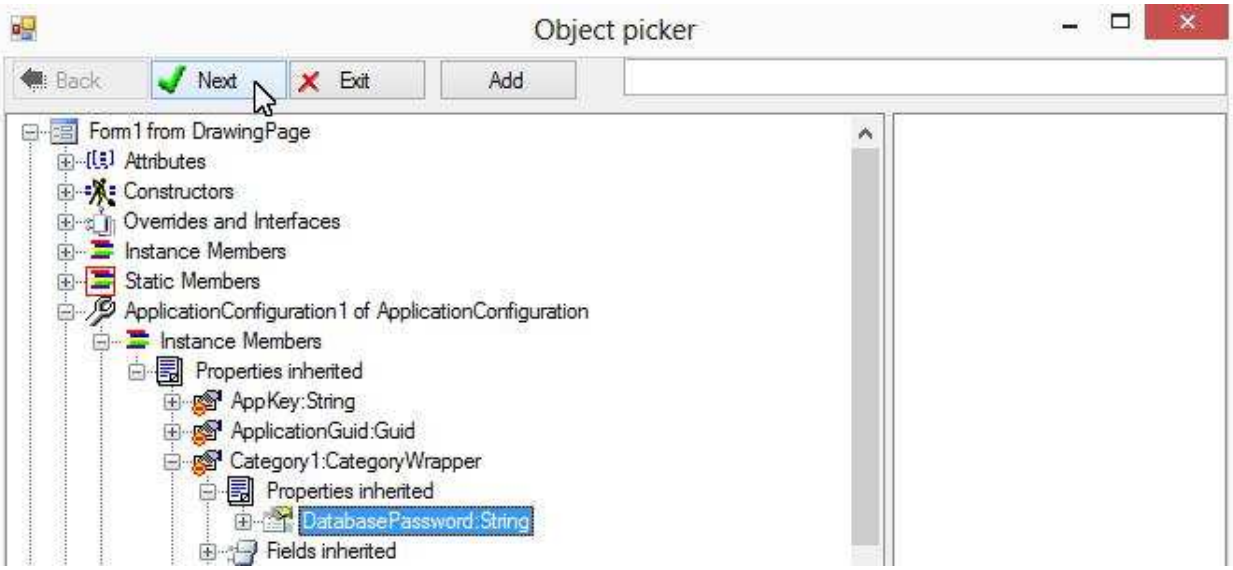


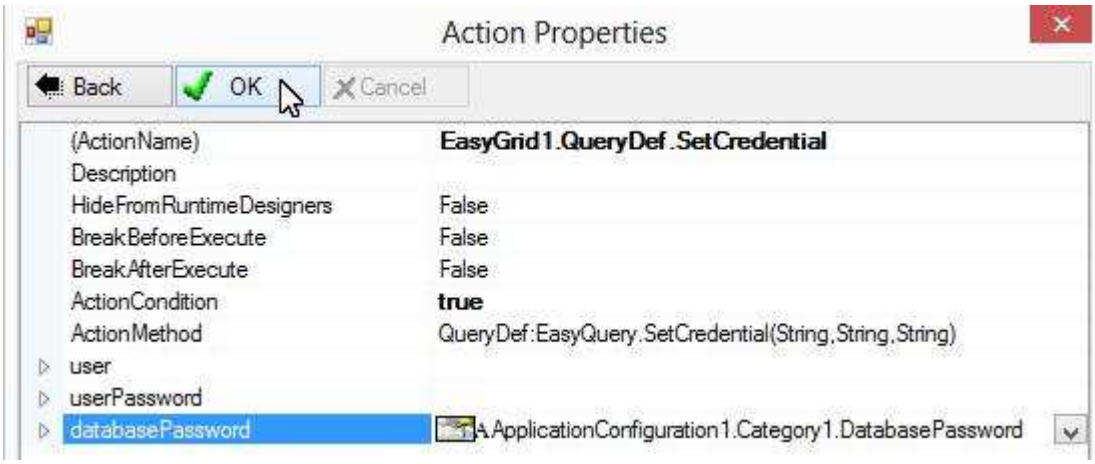
Select SetCredential method of QueryDef:



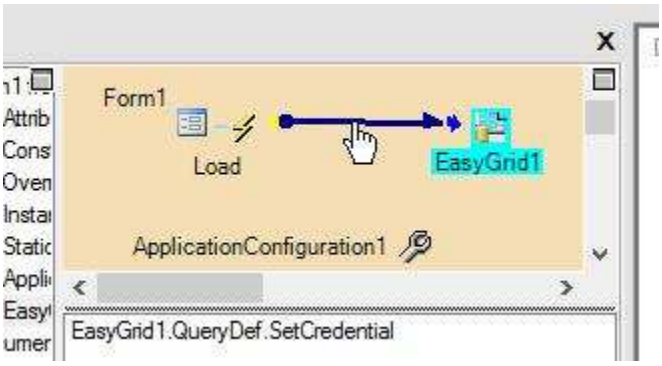
Use database password from the application configuration:



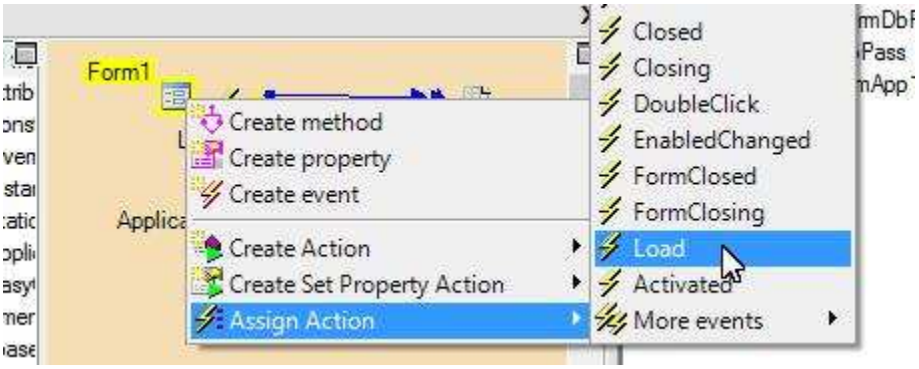


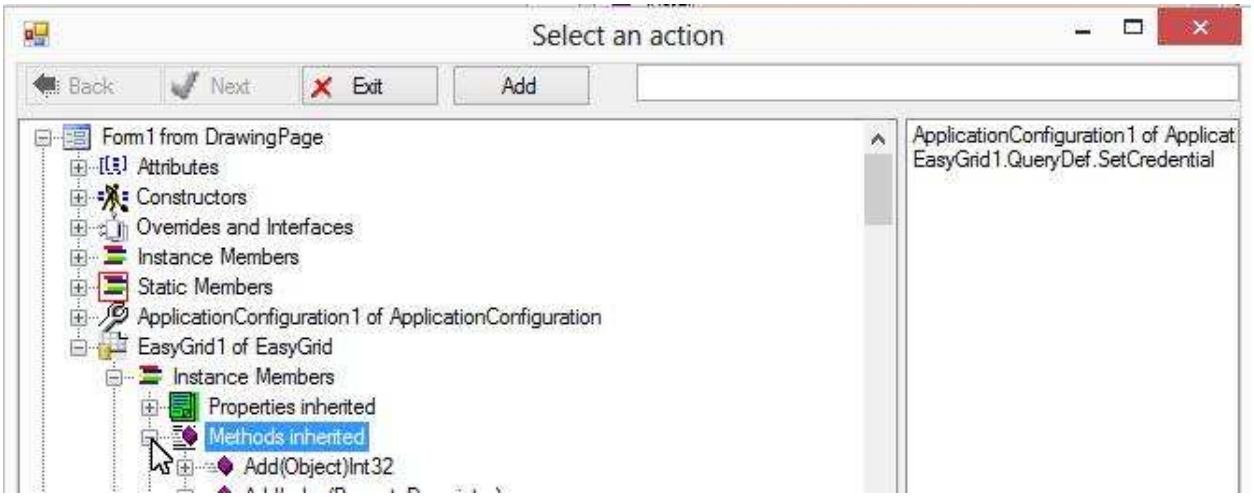


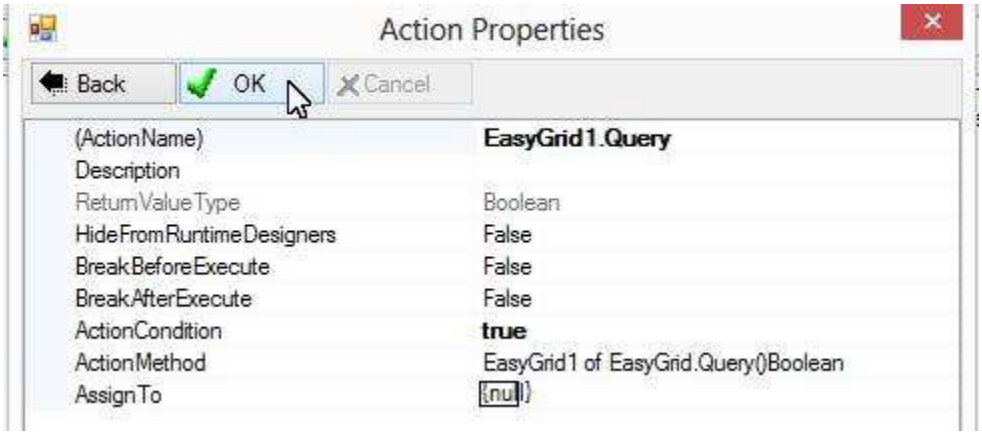
The action is created and assigned to the Load event of the form:



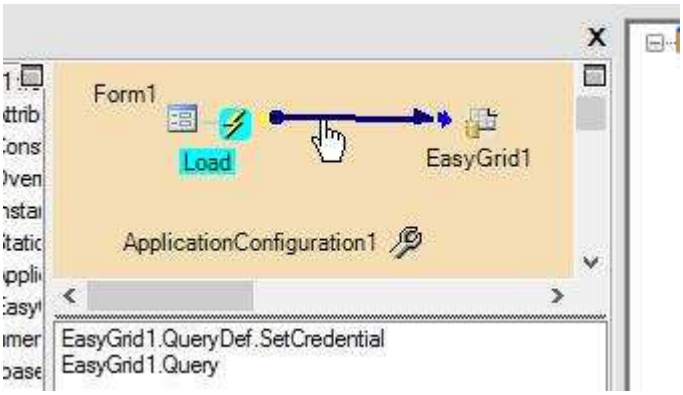
After setting database password, we may do a Query to fetch data from the database:







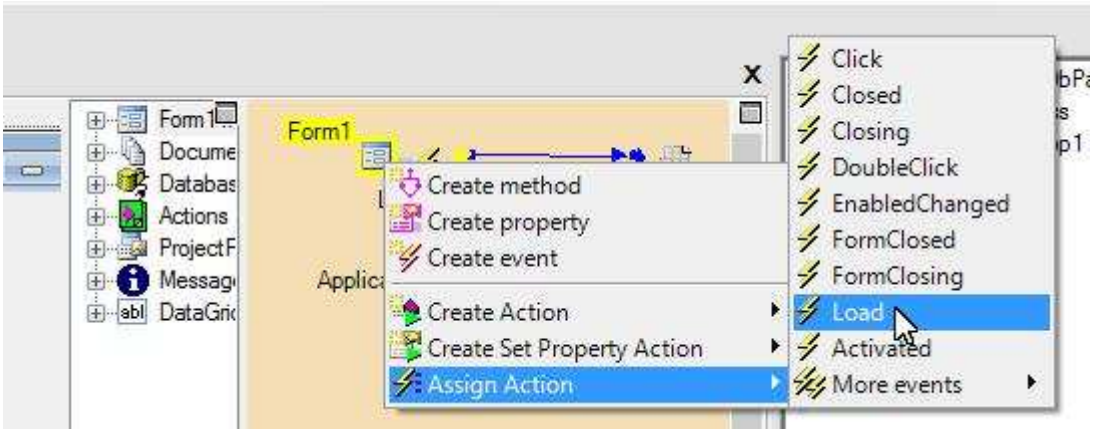
The action is created and assigned to Load event:



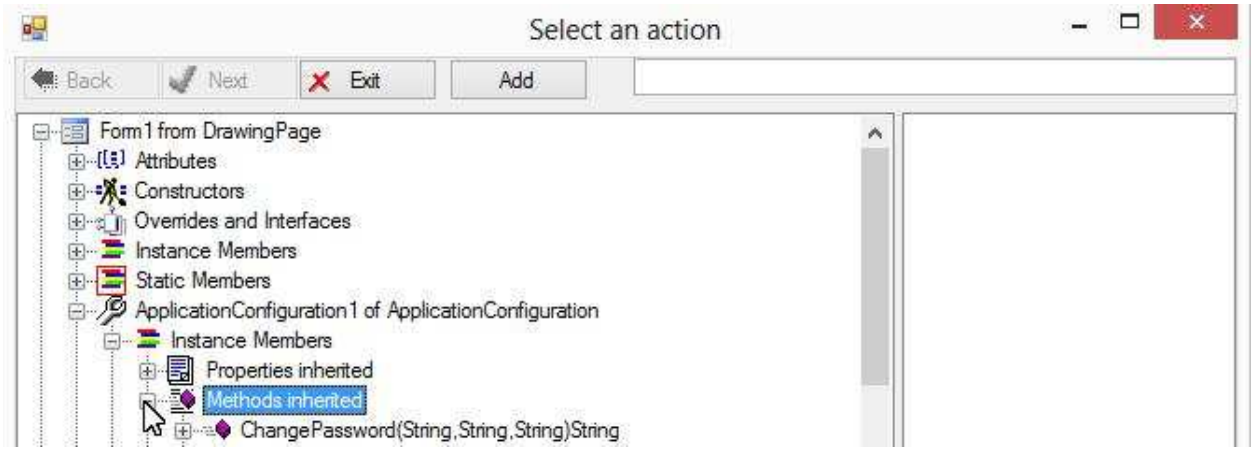
Login to a configuration

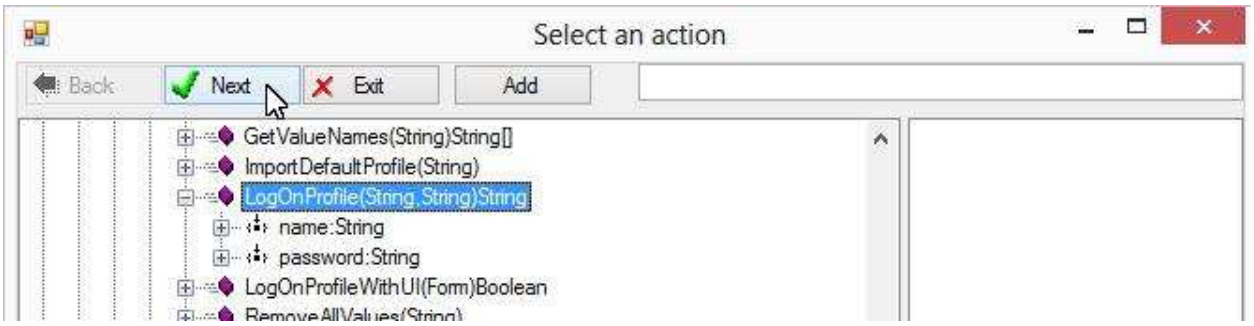
As described previously, the application needs to login to a configuration file before using the configuration file, if the configuration file contains encrypted data.

As described previously, there are options for an application to do login. For simplicity, here we use an automated login for this sample. We add a login action at Load event of the form:

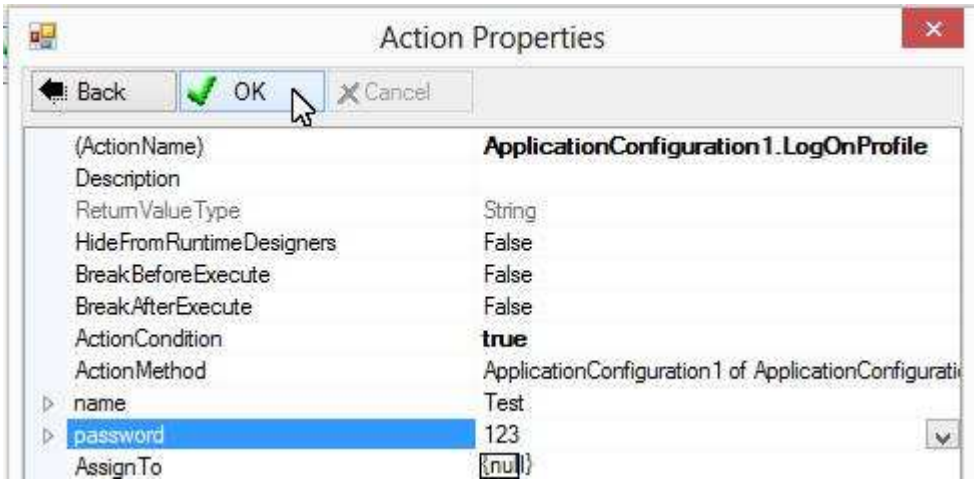


Select Login method of the application configuration:

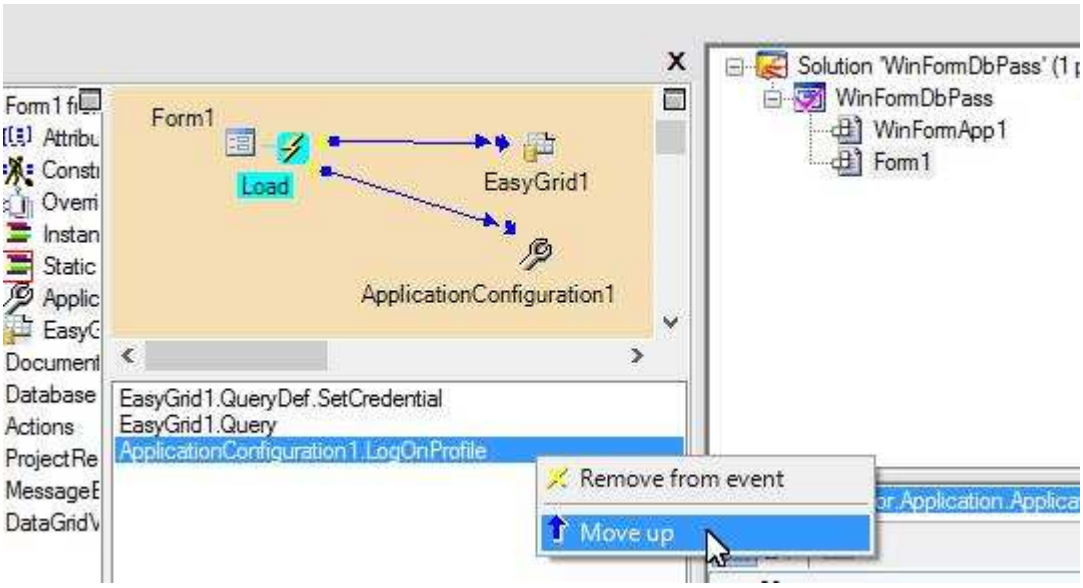


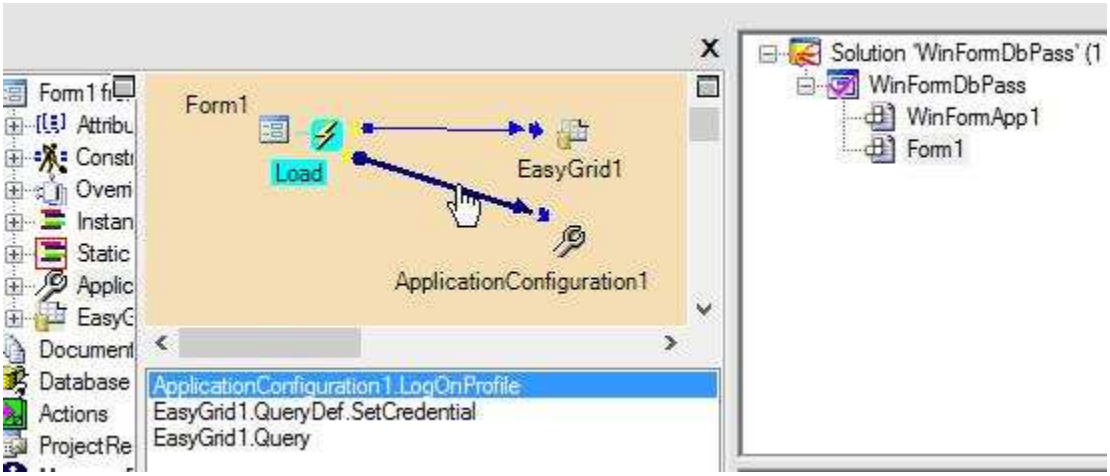


We need to specify configuration name and password for the action. Note that we have not created a configuration file with a name and password. We will do that later. For now, we assumed that the configuration we are going to use is named “Test” and password is “123”:

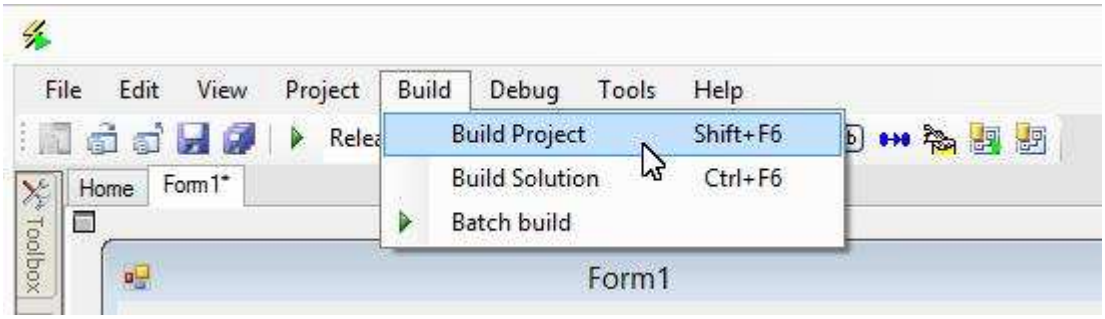


The action is created and assigned to Load event. We need to move it up to make it execute before other actions:





Now we may compile the project to generate EXE file:

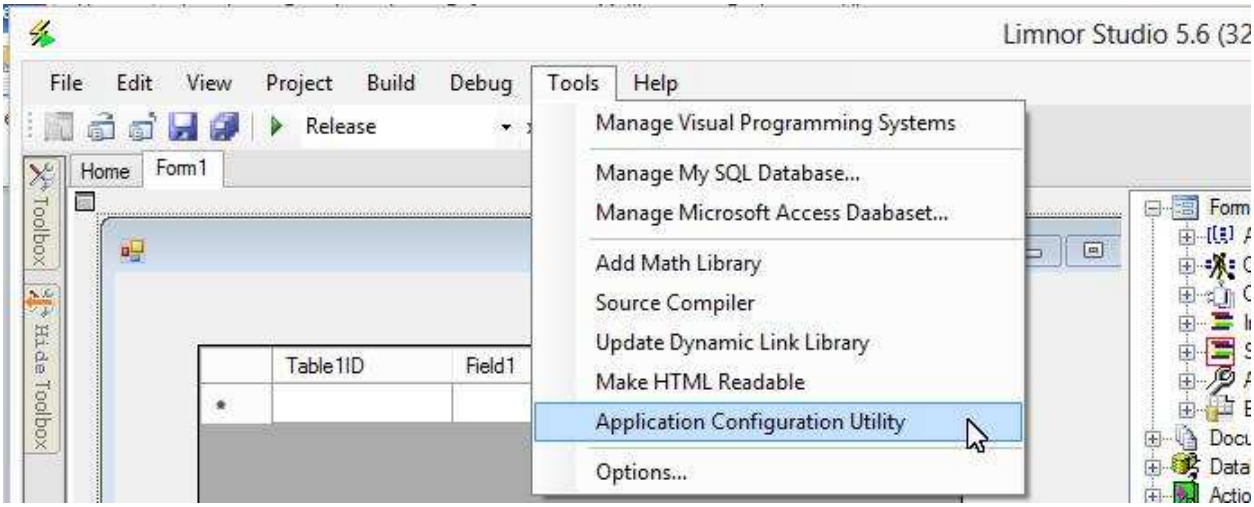


Remember that this application will use a configuration named “Test”. We need to create this configuration now.

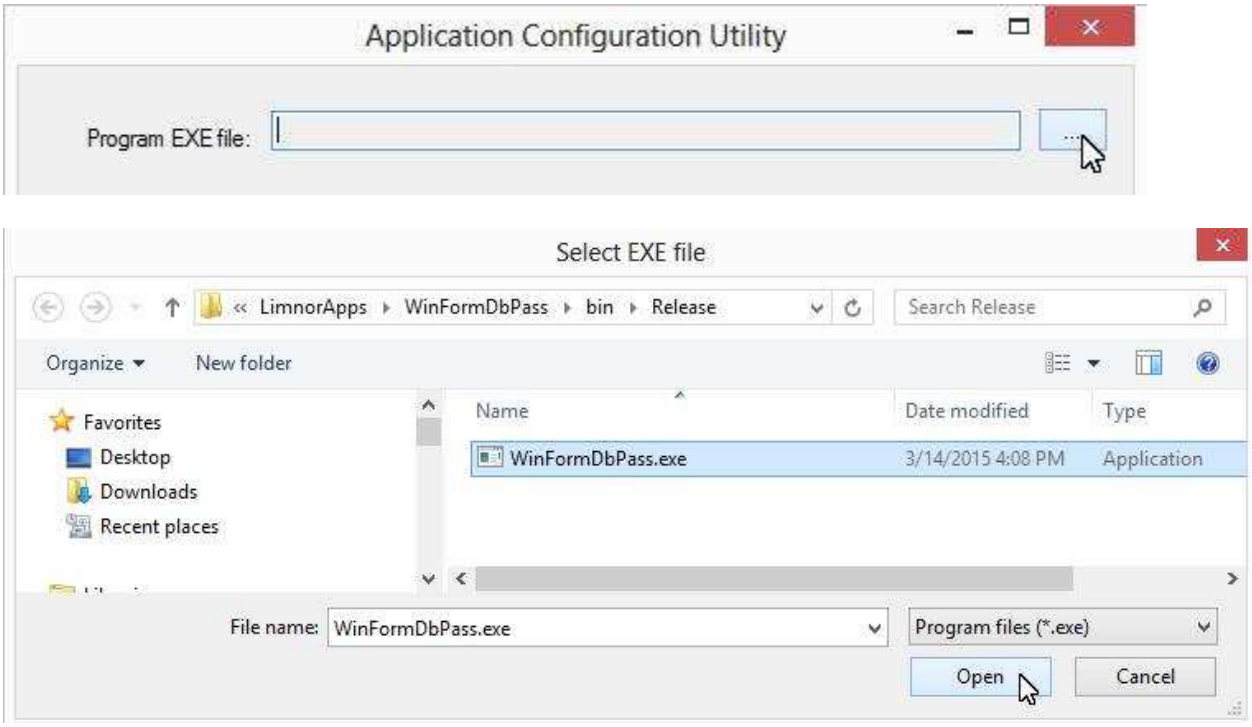
Encrypt password

The technique is already described previously for creating configurations. We duplicate the process for this sample application to create a configuration named “Test” with password “123”.

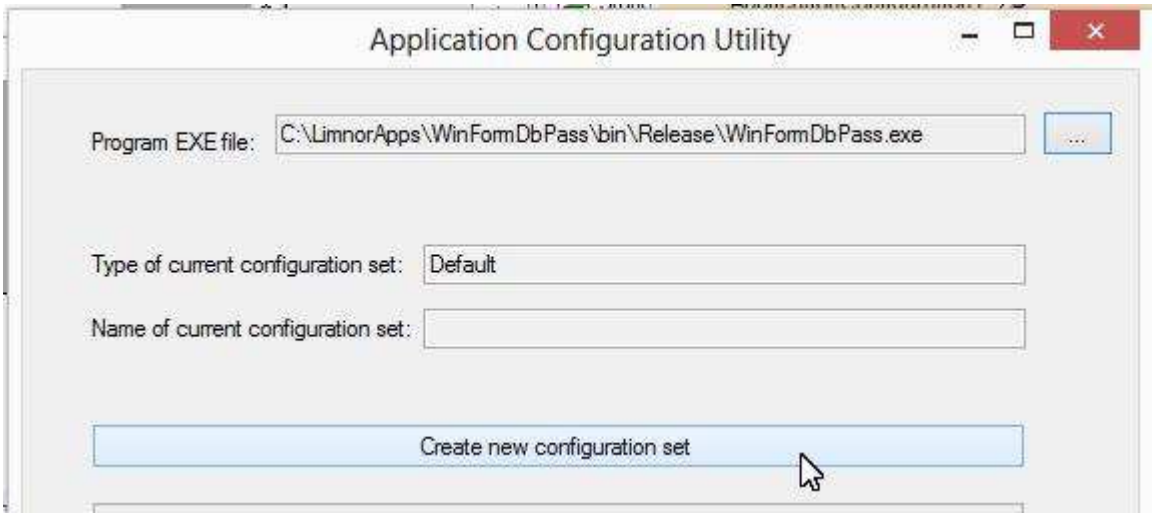
Choose “Tools” and “Application Configuration Utility”:



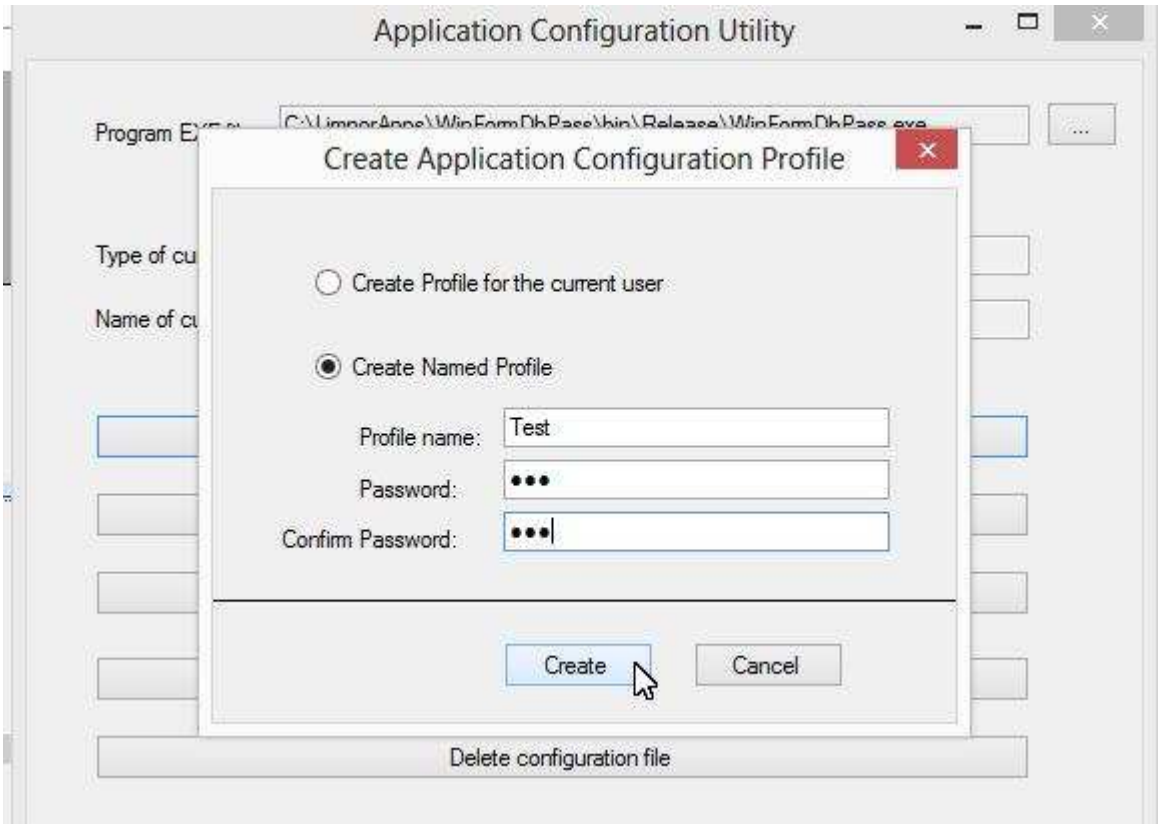
Select the EXE file compiled from this sample project:



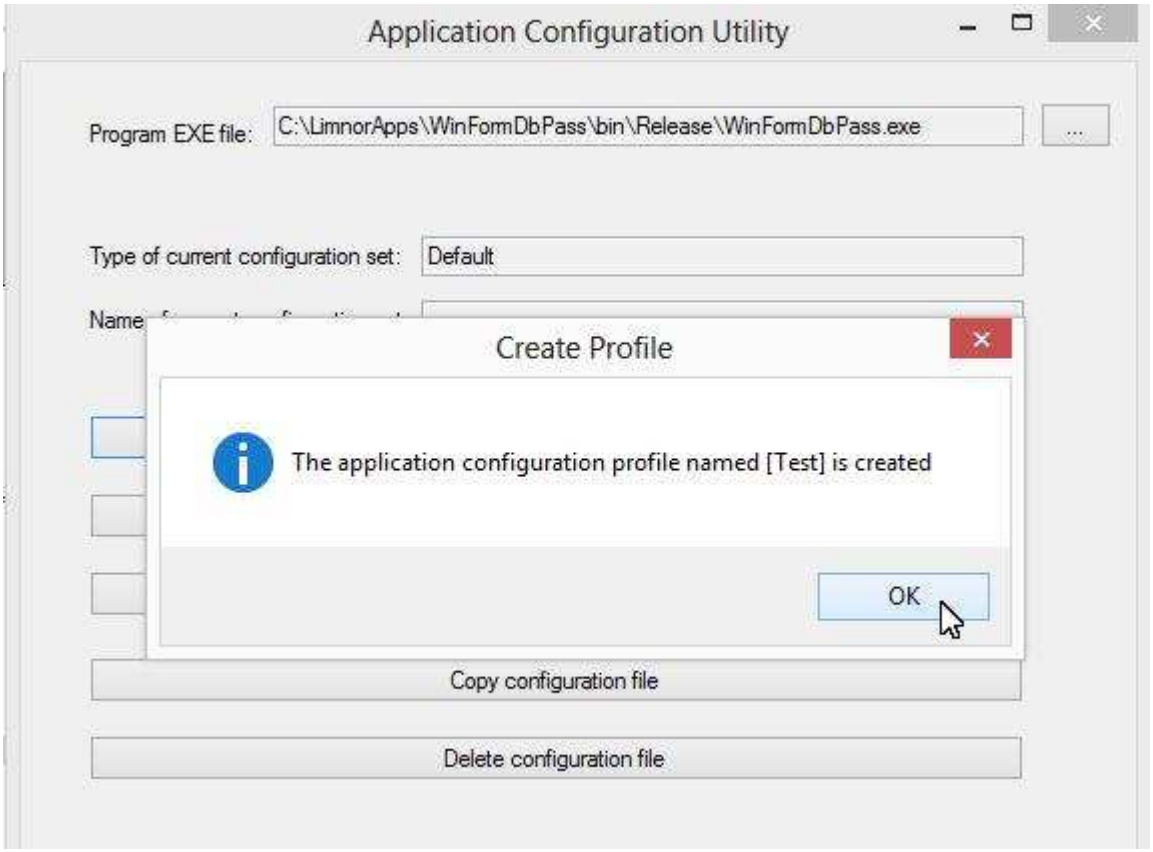
Click “Create new configuration set”:



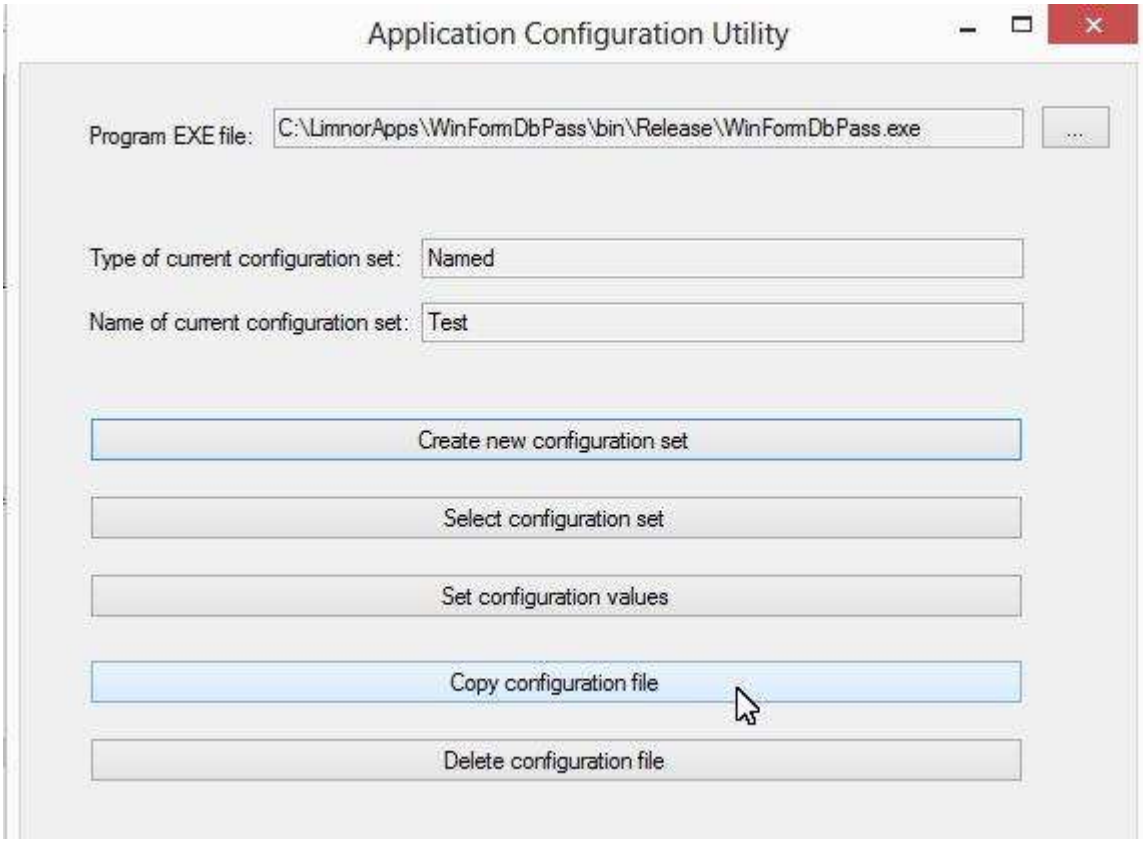
Create a "Named Profile" by giving a name and password, note that this password is not the database password. You select a password to protect this new configuration set. For this sample, the name is "Test" and password is "123":



A new configuration file is created:



Click “Copy configuration file”:



Note that "Current configuration file" shows where the new configuration file is created. "Copy to folder" indicates the folder you use to hold the distribution files for this project. Limnor Studio generates the distribution files under "bin\Release" folder under the project folder. You may manually copy files from "bin\Release" to the folder of "Copy to folder", and then use this utility to copy the configuration file to "Copy to folder". You distribute files in "Copy to folder" to your users.

Feedback

Please send your feedback to support@limnor.com