

User Guide – Use Variables

Variables are memory storages you use to save and retrieve data. You may create variables at both runtime and design time.

[Create Variables at Design Time](#)

[Set Variable Value at Runtime](#)

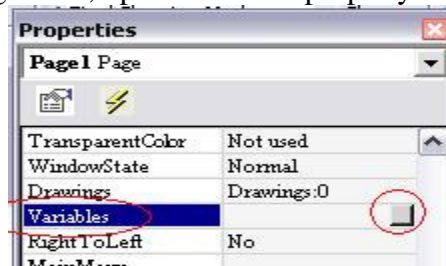
[Retrieve Variable Values](#)

[Create Variables at Runtime](#)

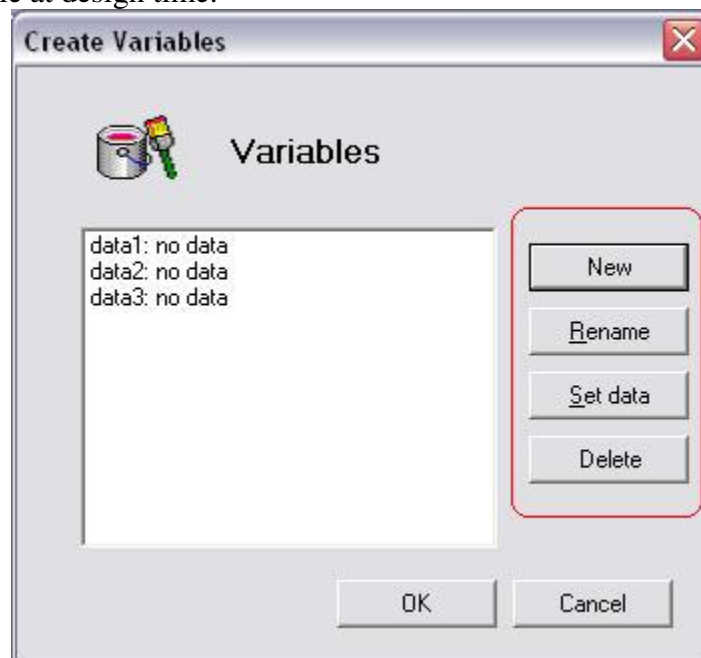
[Scope of Variables](#)

1. Create Variables at Design Time

Usually, variables are saved in a property named **Variables**. Many kinds of performers have this Variables property, for example, Application, Page, Math Expression, etc. To create variables at design time, open Variables property:



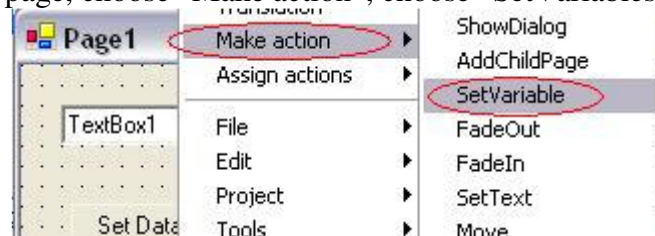
A set of buttons are provided for managing the variables. New button is for creating a new variable. Rename button is for changing the variable name. You give a unique name to identify each variable in Variables property. “Set Variables” button is for assigning value to a variable at design time.



2. Set Variable Value at Runtime

Usually a performer with a Variables property has a method named **SetVariables**. This method has two parameters. The first parameter is the name of the variable. The second parameter is the value to be assigned to the variable. You use SetVariables to create actions to set variable values.

For example, suppose we have a text box performer, and we want to assign the Text property of the text box to a variable named Data1. We may create an action to do that. Right-click on the page, choose “Make action”, choose “SetVariables”:

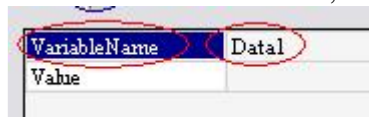


Give an action name, say, Page1.SetData1:

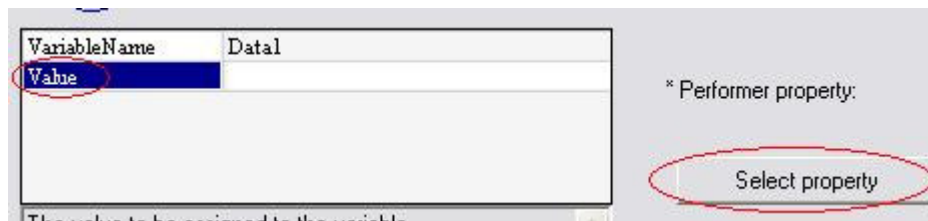


The Action Data dialogue box appears.

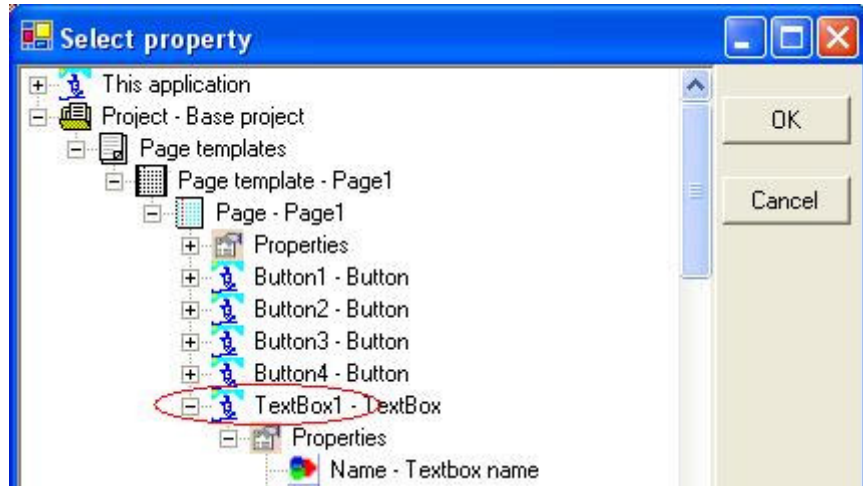
For VariablesName, it is the variable name. In this case, it is Data1:



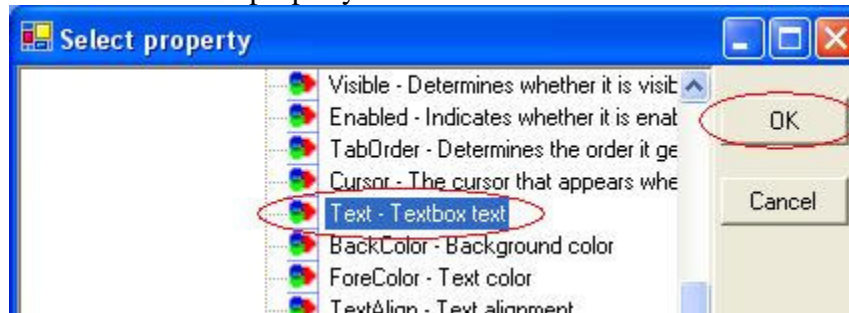
For Value, we want to use the Text property of the text box, so click “Select property” button:



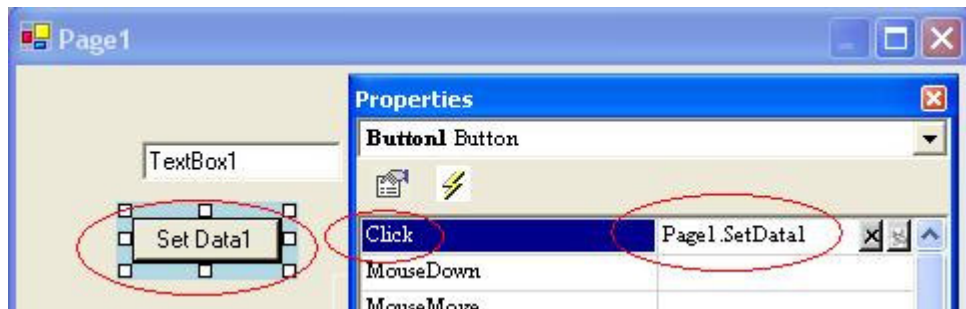
Find the text box performer:



Scroll down and find the Text property:



Finish creating this action and we have created a new action named Page1.SetData1. Assign this action to the Click event of a button:

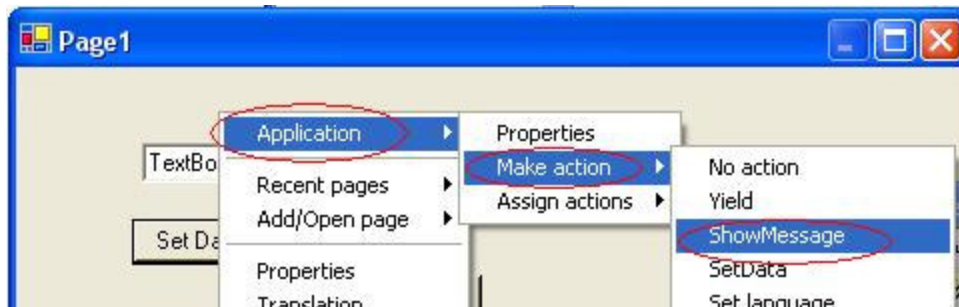


At runtime, clicking this button, whatever the user types in the text box will be copied to variable Data1.

3. Retrieve Variable Values

Variables are used the same way as all other properties are used. You may use variables as action parameters.

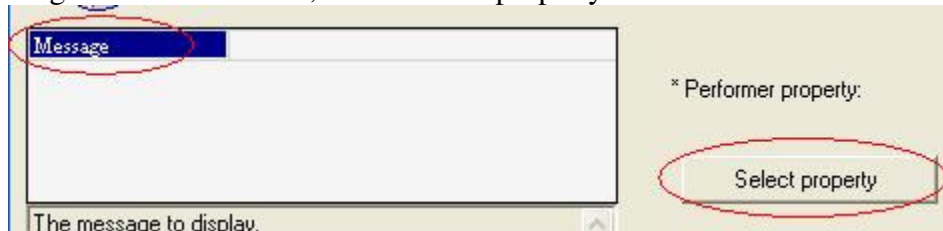
For example, the Application performer has a method named ShowMessage which shows a message box. It has one parameter which is the message to be shown. Let's use it to create an action to show the value of variable named Data1. Right-click on a page, choose "Application", choose "Make action", and choose "ShowMessage":



Give an action name, says, Application.ShowData1:



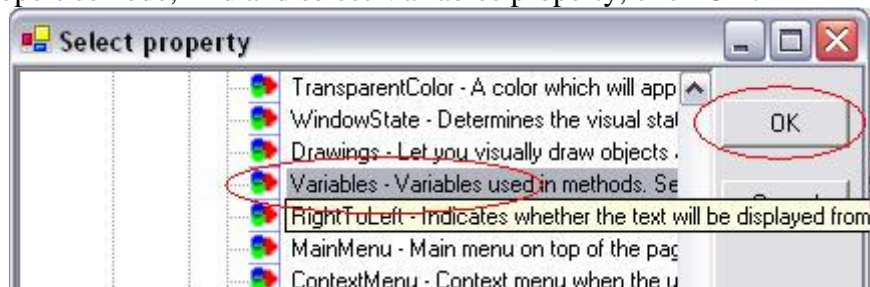
For showing the value of Data1, click “Select property” button:



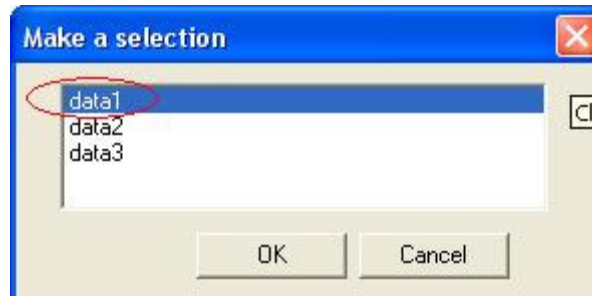
Find the page containing the variables:



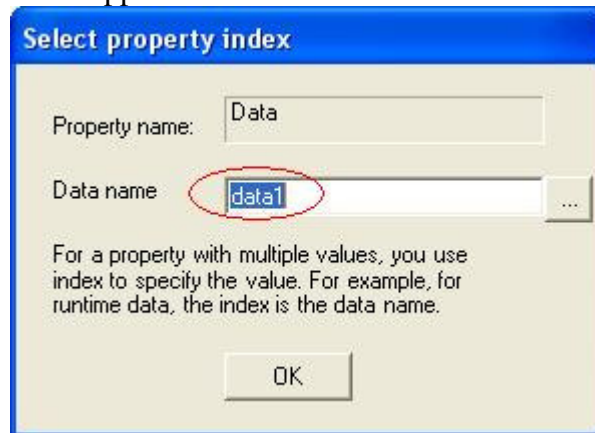
Expand Properties node, find and select Variables property, click OK:



You may click button  to pick the variable by name:

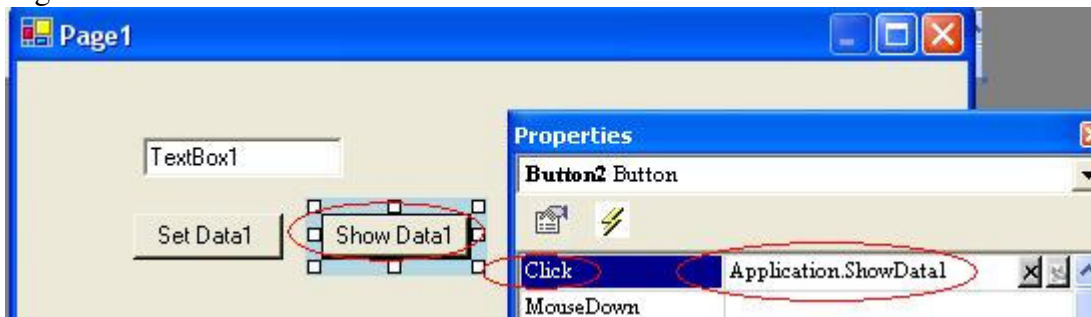


The selected variable name appears in Data name box:



Finish creating this action and we have created a new action named Application.ShowData1.

Assign this action to a button's Click event:



Now run the application. Type in something in the text box, click button “Set Data1”, click button “Show Data1”, you will see a message box showing whatever you type.

4. Create Variables at Runtime

At runtime if SetVariables is called, see [section 2](#), and a variable does not exist for the variable name, a new variable will automatically be created.

5. Scope of Variables

You, as a developer, need to be aware of when variables are available, and thus you will be using the values you expected. If a variable is saved in the Variables property of a

page, or on the Variables property of a performer which belongs to a page, your application should use the variable while the page is loaded in memory.

Suppose you create a variable named MyValue1 on a page named Page1 at design time, and assign a value 12 to MyValue1. If you use Show or ShowDialog method to show Page1 then variable MyValue1 will be available and has a value of 12.

Now suppose an action sets the value of MyValue1 to 16. You will expect all other actions using MyValue1 will get value 16.

Now suppose Close method is called for Page1. Page1 will be removed from memory. MyValue1 will be gone with Page1 (out of scope).

At this time if an action is executed and using MyValue1, and if you expect to get a value of 16, your application will not be acting as you expected because MyValue1 will have a value of 12, not 16 as you expected. What happens here is that when accessing MyValue1, because Page1 is not in the memory, Page1 is reloaded into memory, and MyValue1 will have its initial value set at design time, that is 12, not 16.

If a page is not in memory when its properties are accessed, or a performer on it is accessed, the page is automatically loaded into memory, with all its properties and the properties of performers on it having values set at design time.

If you are sure a variable will only be used while a page is in memory then you may put this variable in the Variables property of the page. In this way when the page is closed, the variable is also removed from memory.

If you want a variable to be available all the time while your application is running, you may put the variable in the Variables property of the Application performer, or the Variables property of a performer under the Application.