Users' Guide for Beginners –Part I

Contents

Introduction

Our vision is that everyone should be able to do computer programming. Our approach is to do computer programming visually and codelessly. This users' guide uses as few technical terms as possible. When a technique term is used it is explained to avoid misunderstanding.

Object-based programming

A computer language is formed by keywords (if, else, …), symbols (+, -, …) and syntax (grammar).

To a beginner the term "computer language" might have magic power to instruct any computers to do anything by a language itself. It is not literally true.

- ✓ Different computers built with different hardware have different functionality. Some early computer manufacturers provided computer languages shipped with their computers and added specific functionality into their computer languages to match their computer functionality. For example there are many flavors of BASIC languages. A user learned such a computer language might be confused that the same language does not work in other types of computers.
- ✓ New hardware may be plugged into an existing computer. The hardware manufacturer will provide software library to allow computer languages to access the new hardware functionality. Such software library is called "Software Development Kit" (SDK); it is not part of computer languages.

Actually for more advanced computer languages most of functionality is not expressed by keywords. All functionality is in software libraries. For example, suppose we want to show a text in the screen, in BASIC language we can find a keyword PRINT to do it. But in C language we cannot find a keyword such as "PRINT" or "ShowText" to do it. Instead in the standard input and output software library there is a function "puts" and a function "printf" which can be used to show text in the screen. You may ask why I should care about distinguishing between computer language keywords and function names. Well, you don't, if you use codeless programming.

Software libraries for modern computer languages, such as C++, classify functionalities into objects. For example, a Console object may contain all functionalities related to a computer console, such as showing text, getting keyboard inputs, etc. An FTP Client object may contain functionalities related to downloading and uploading files to and from FTP servers.

In such object-based paradigm, to look for a specific functionality, we do not look for it in the language itself. We try to find an object in a software library to provide the functionality we want. If we cannot find the exact functionality we want then we find an object which provides the closest functionality to what we want. Using the found object to create a new object and add new functionality to the new object while taking advantages of the functionality provided by the found object.

If you are not used to object-based programming then you may be frustrated by the wide range of functionality spreading among so many objects in so many software libraries. Just note that it is to your advantage that a vast number of software libraries are available for you to take advantage of. New software libraries are added to the market constantly. You may also create your own software libraries.

You may search in the Internet for objects you want, or ask in various forums for your task requests.

Codeless programming

In object-based programming, a language acts as glue to link objects together to form new software, or as nails and rivets to link building blocks together.

It is also like using Lego blocks to form constructions. But Lego constructions do not need glues, nails and rivets. It is because that each Lego block is made with pins and sockets to be interlocked to other Lego blocks.

Modern software objects are also made with pins and sockets to be interlocked to other objects. Therefore a language is not needed to glue objects together. That is the idea of codeless programming. Limnor Studio is an experiment on this idea. Technologies used in Limnor Studio are patented.

Using a few simple mouse and keyboard operations, you interlock objects together to build your software.

This tutorial explains how to use Limnor Studio to link software objects together to form new software. There is not a textural computer language to learn. Just manipulate objects visually.

Object as Programming Entity

What is Object

We see the world by objects. A tree is an object. A house is an object. A TV is an object. In object-base programming, we also see everything as objects. A number is an object, a button is an object, a window is an object, etc. We are always dealing with objects.

Technically speaking, objects in object-based programming are defined by 3 kinds of attributes: Properties, Methods, and Events.
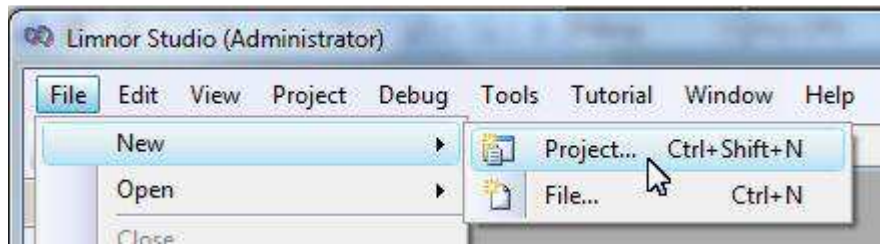
To understand an object's functionality is to understand its properties, methods, and events.

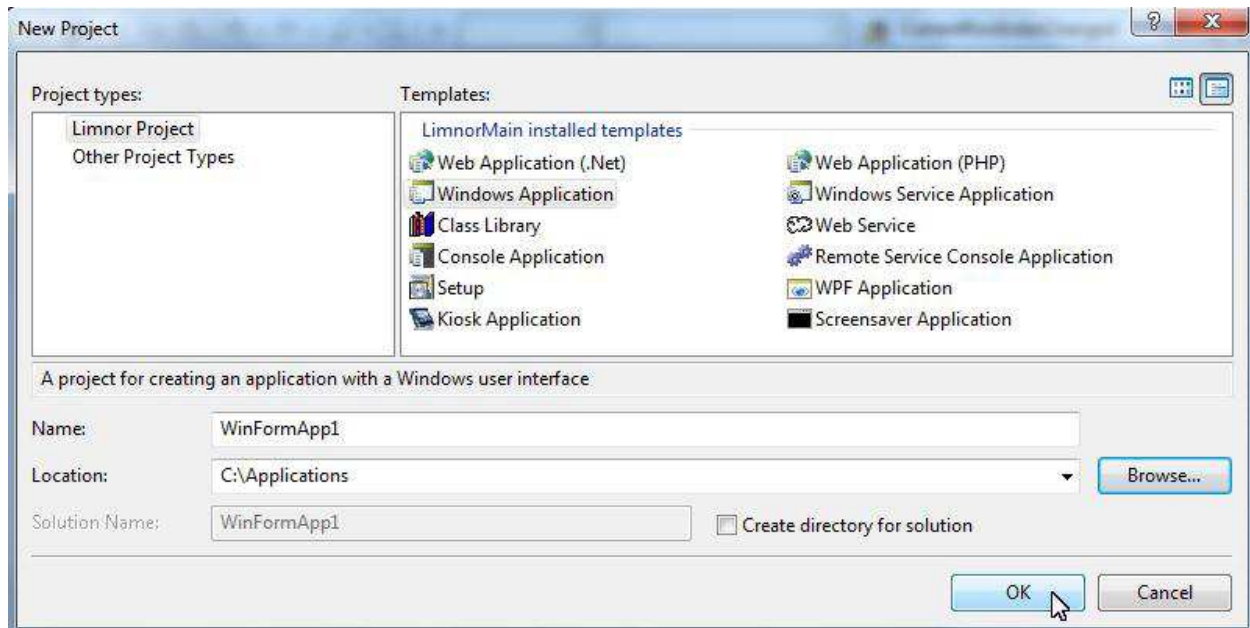To do object-based programming is to create new objects.

Create Objects

We will explain properties, methods and events using examples. To get object examples, let's create a Windows Form application.
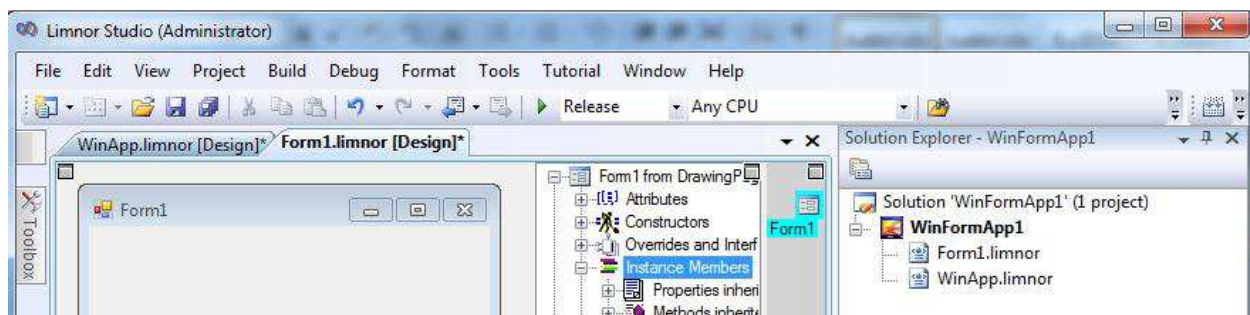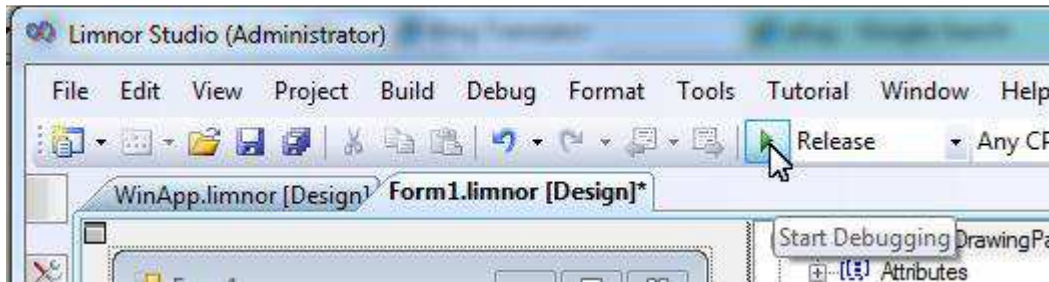
Choose menu "File|New|Project"
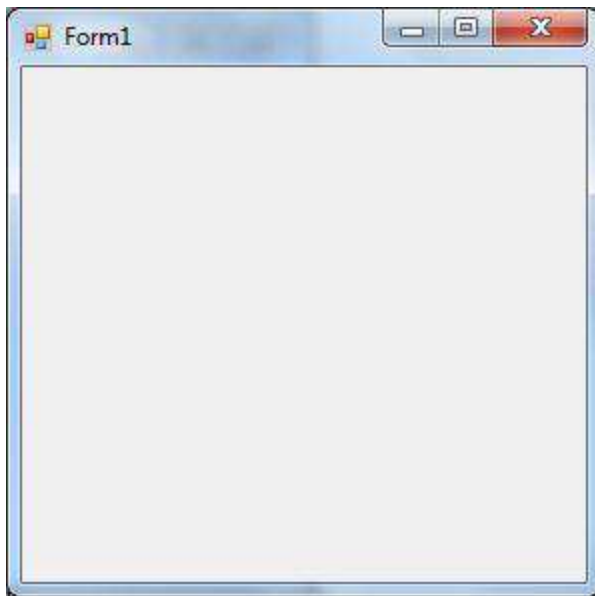
Choose "Windows Application"

A Windows Form project is created. Two objects are created for this project. An object named WinApp represents the Windows Form application. An object named Form1 represents a window.

Click the Run button to test this application:

This application runs. A window appears:

This window is object Form1. The object WinApp is not visible because it does not have a visible user interface. Actually most objects are not visible when an application runs; you only saw them in Limnor Studio when you are developing your project.
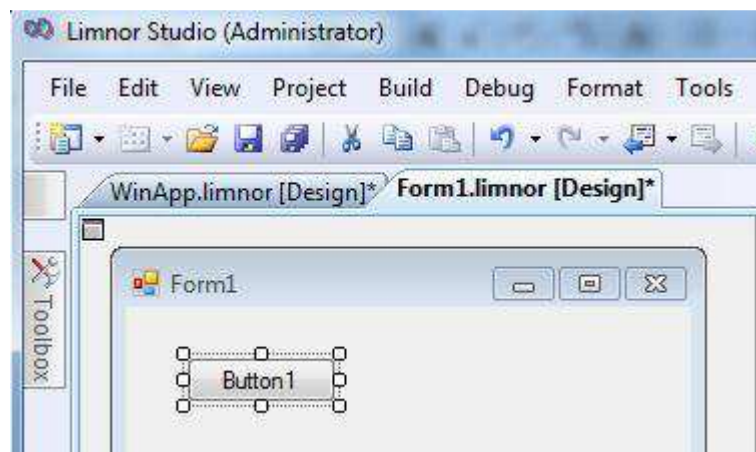
Close the window Form1, the application exits because Form1 is the first window of the application. To resume designing the object we must exit the application.

We may use Limnor Studio to visually develop objects.

For example we may drop a button object from the Toolbox to Form1:
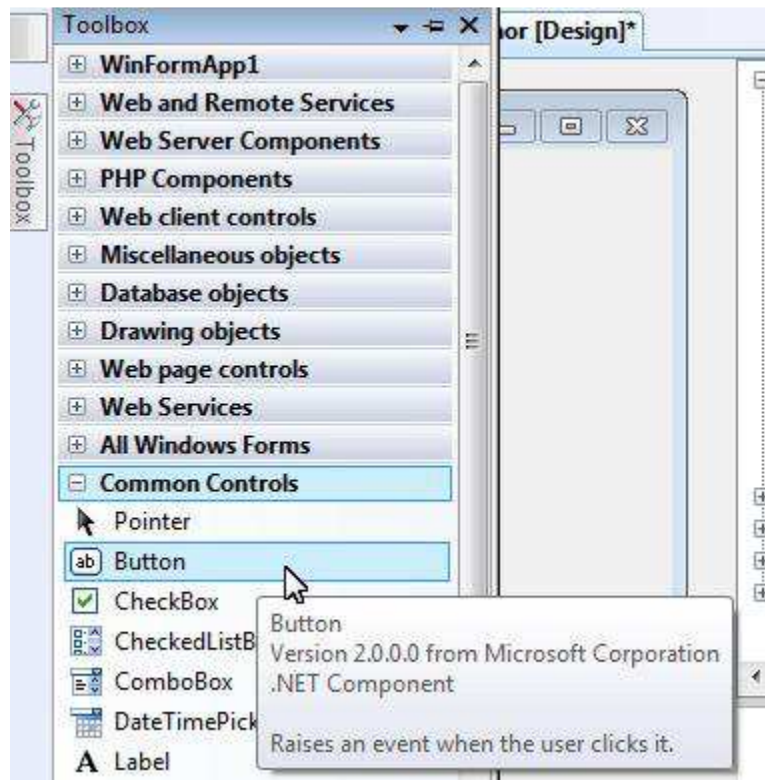
The button appears on the form:

The button object, Button1, and the form object, Form1, are linked together, just like two Lego pieces interlocked together.
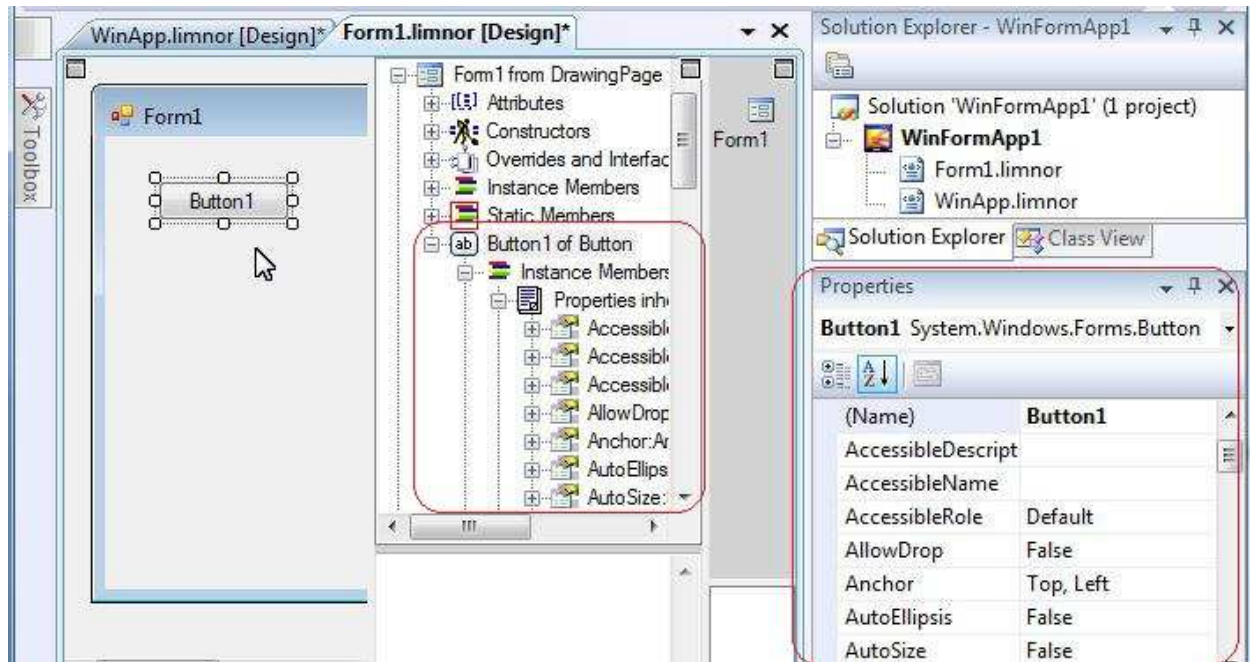
Properties

Suppose we want to know how tall a tree is, then the height of a tree is a property of the tree object.

A property of an object is a value associated with the object and representing one aspect or characteristic of the object.

When an object is selected, its properties are displayed in a Properties window and in the Object Explorer:

For example, the Text property of a button object represents the caption of the button. Set Text property of Button1 to "Close", we can see it appears on the surface of the button:
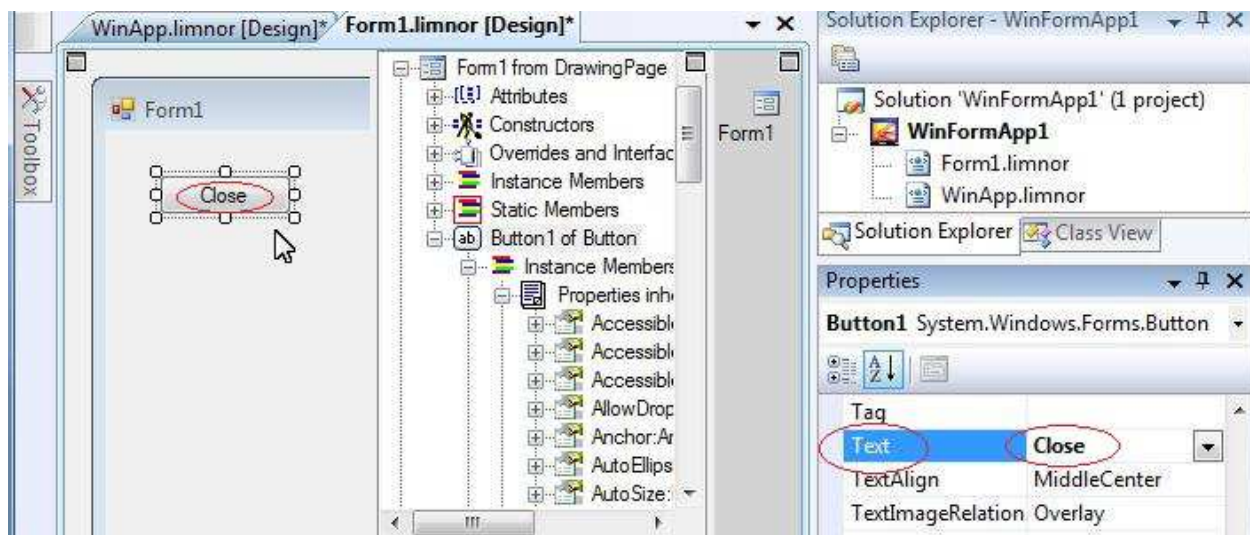
A button object has other useful properties, such as Image, Location, etc.

For documentation of all properties of Form object, see http://msdn.microsoft.com/en-us/library/system.windows.forms.form_properties.aspx

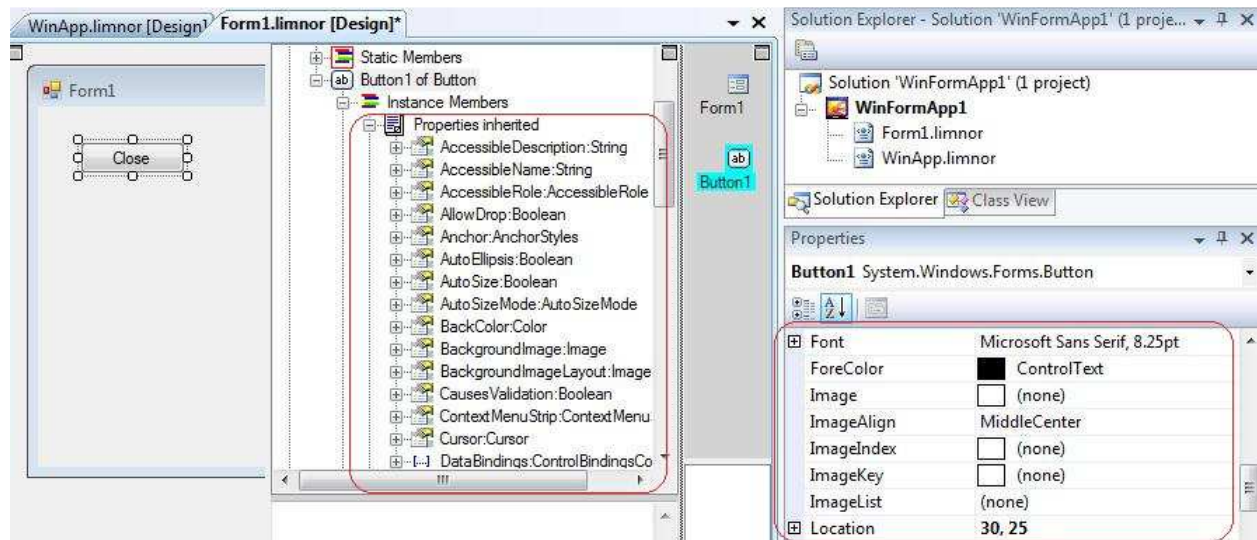For documentation of all properties of Button object, see http://msdn.microsoft.com/en-us/library/system.windows.forms.button_properties.aspx

Keep in mind that learning the properties of objects is not part of learning Limnor Studio and codeless programming. Objects belong to software libraries, not Limnor Studio. In Limnor Studio, you may modify the object properties visually and your work will be saved to files.

In Limnor Studio, every object is given a name to uniquely identify the object. You may change the names of objects.

In above example, we modified object Form1 by adding a button object to it. We modified the caption of the button on Form1. These operations are part of programming.

Methods

A method of an object represents one thing the object can do.

For example a singer can sing songs. "Sing song" is one method of a singer object.

The concept of "method" is closely related to a concept of "action". An action is to do a specific thing. For example, "Sing 'Silent Night'" is an action which is "Sing Song" but it is to sing a specific song.

A "method" indicates an ability to do something; an "action" is really doing the thing.

All methods of an object can be found in the Object Explorer:
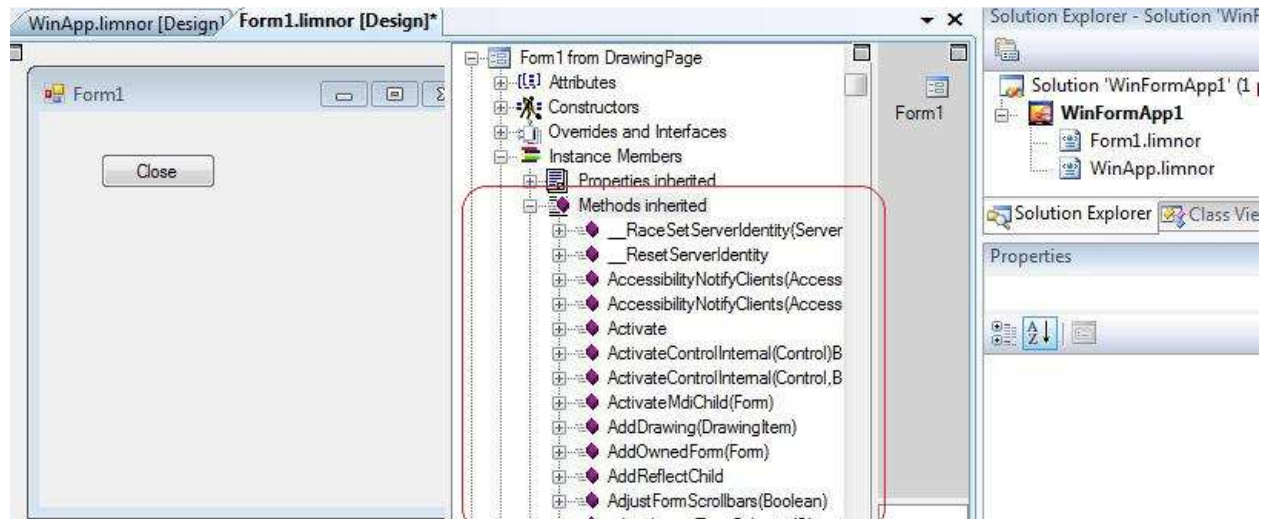
Actions can be created using the methods. For example, right-click a method in the Object Explorer, choose "Create action":
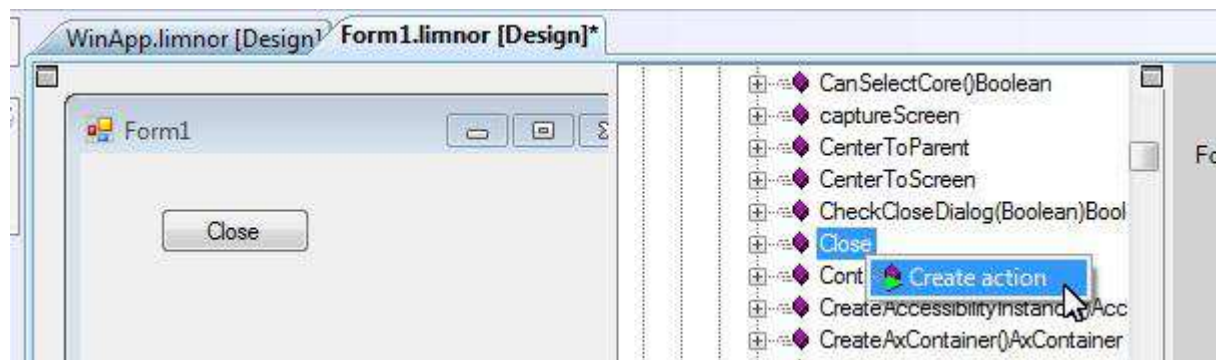
Another easier way to create an action is by right-clicking an object, choose "Create Action", and then choose the desired method. For example, right-click Form1 object, choose "Create Action", choose "Close" method to create a "Close" action:

In object-based programming, everything is an object. An action is also an object. When we create an action, a dialogue box appears to allow us to specify the properties of the action.

You may give the action a name. You may set "ActionCondition" so that the action can be executed only if certain conditions are met.

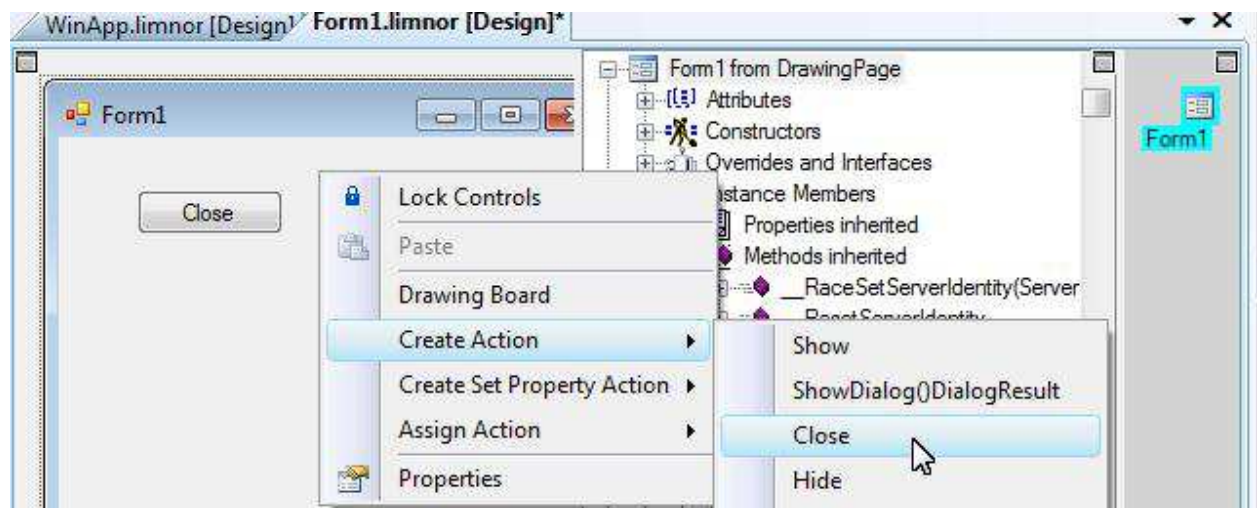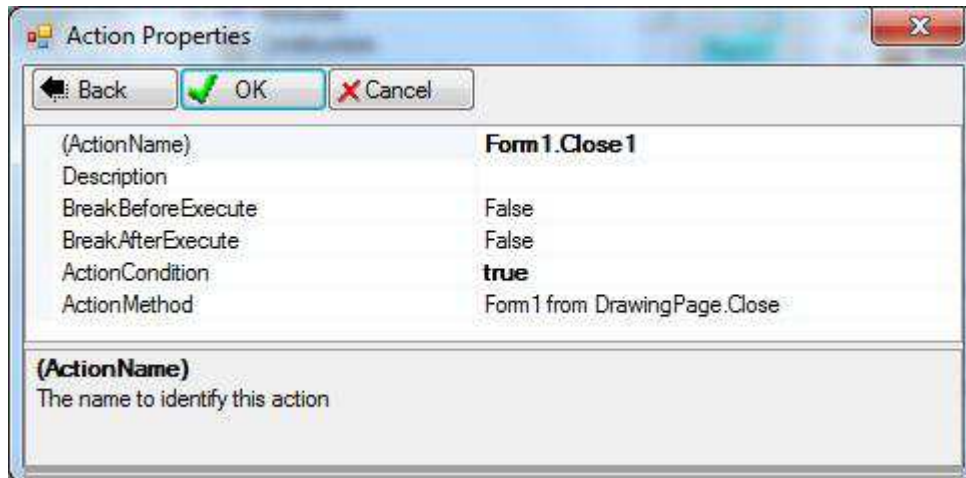If the method for the action requires parameters then the parameters become properties of the action for you to specify the values. For this "Close" method it does not require parameters.

The actions created can be found in the Object Explorer. Select an action, like selecting other objects, its properties are displayed in the Properties window for you to view and modify them.
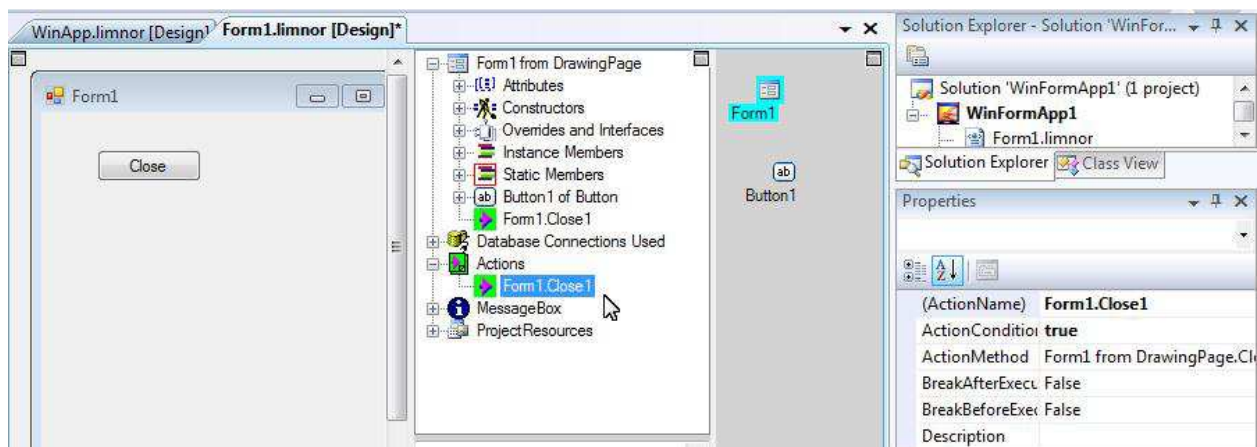
We created the "Close" action. But when should we execute the action?

This question is answered by concept of "events".

Events

An event is one specific moment occurs to an object. For example, we may say "take off" is an event of an airplane, which occurs when the airplane leaves the ground.

All events of an object can be found in the Object Explorer.
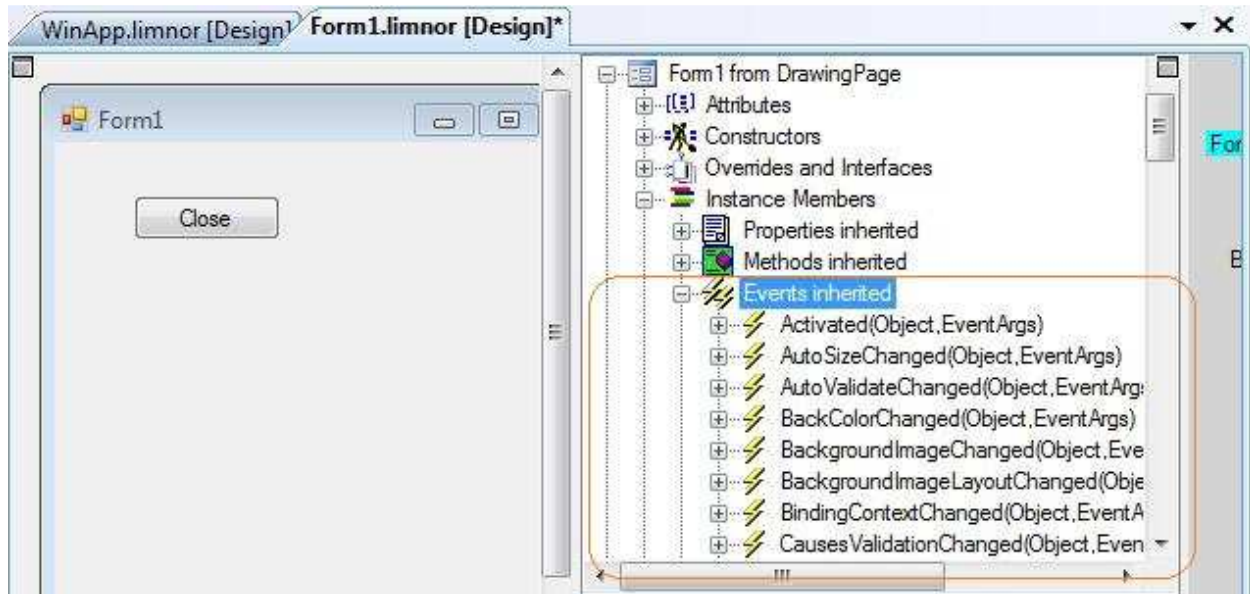
In Limnor Studio codeless programming, we assign actions to events. When the software executes, the actions are executed at the moments when related events occur.

We have created a "Close" action in the last section. Suppose we want the action to be executed when the user clicks the button. In object-based programming, we say that we want to execute the action when the Click event of the button occurs.

To do this programming in Limnor Studio, right-click the button; choose "Assign Action"; choose "Click" event:
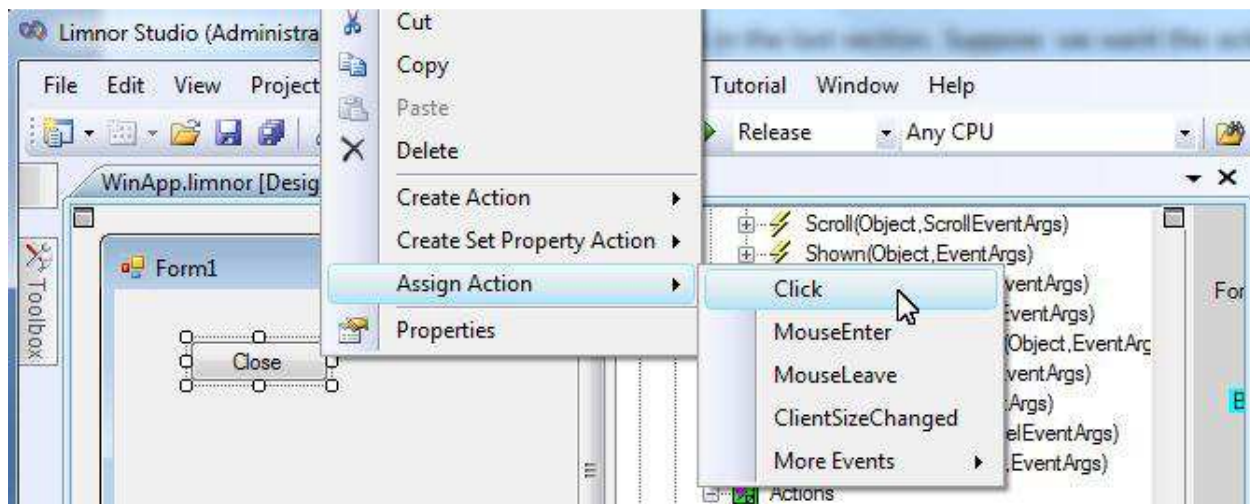
A dialogue box appears to let you select actions. More than one action can be selected. For this sample, we select the "Close" action and click "Next":
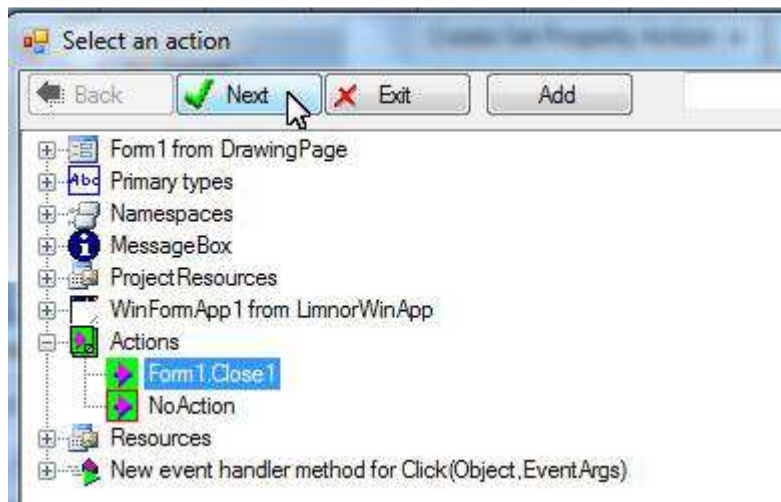
The action is assigned to the event. We can see that the action appears under the event in Object Explorer:
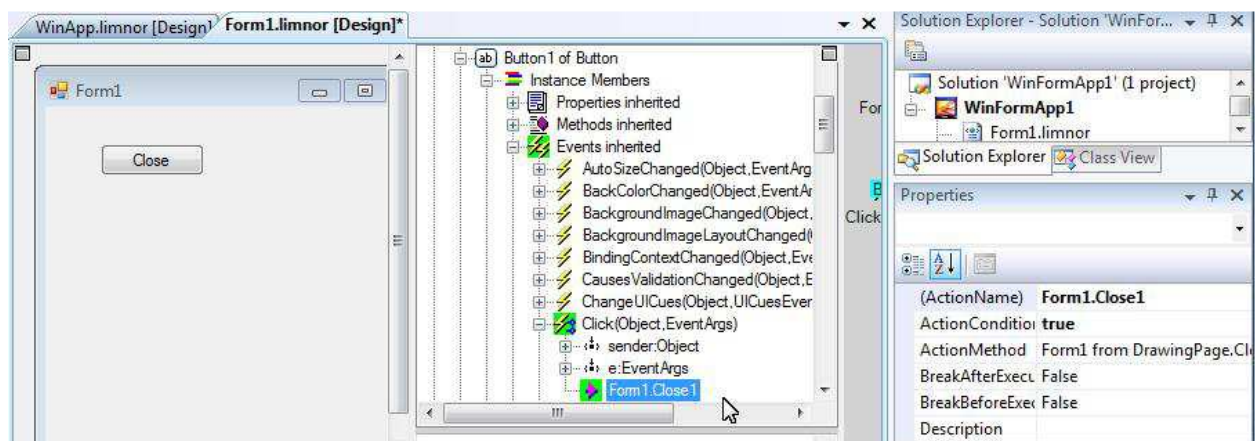
Event Map shows actions-events assignments in diagram:
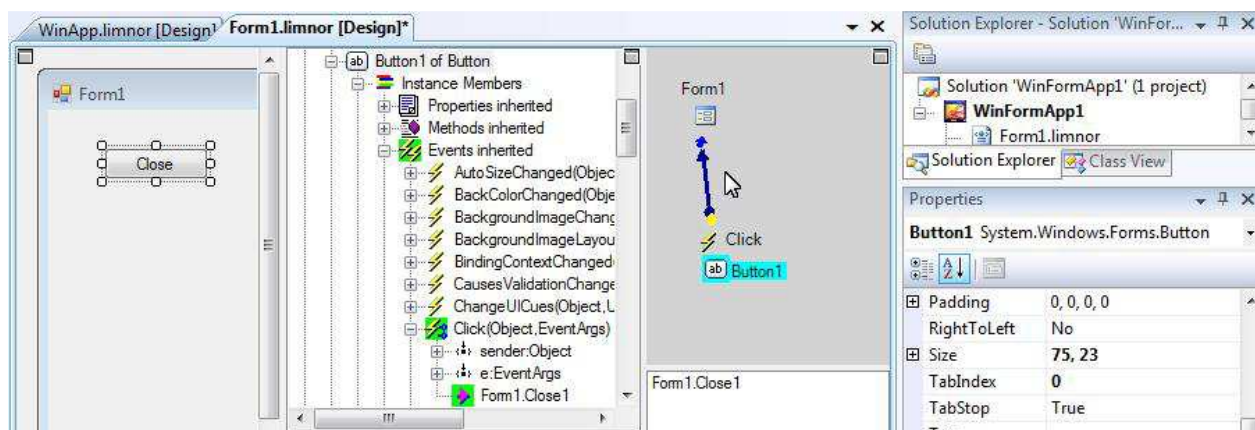
We can see that a line goes from the Click event of Button1 to Form1. This line indicates that when the Click event of Button1 occurs, Form1 will execute one or more actions. Select the line, the actions to be executed will be listed at the bottom of the Event Map. In our example, only one action, Form1.Close1, is listed.

Our programming is done. We may test our software now. Click the Run button:
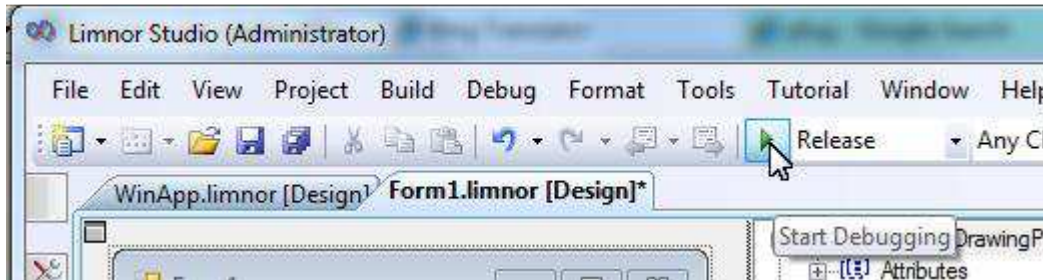
Form1 appears.
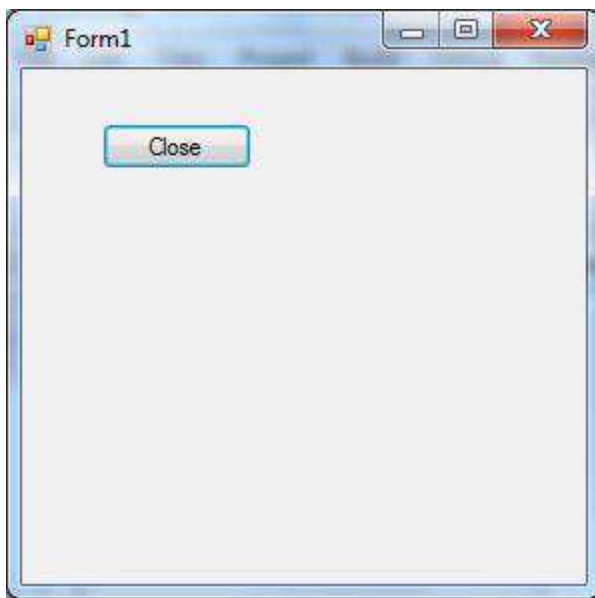
Click the button, Form1 disappears and the application exits, as the result of executing action Form1.Close1.

The programming we just did is as following: object Form1 has a "Close" method to close itself; we used the "Close" method to create an action named Form1.Close1; we assigned the action Form1.Close1 to the "Click" event of the object Button1.

The result of the programming: click the button, Button1, and the window, Form1, closes and the application, WinApp, exits.

Using our "Singer" metaphor, object "Singer1" has a "Sing Song" method to sing a song; we use the "Sing Song" method to create an action "Singer1.SingSong1" to sing "Silent Night"; we assign the action "Singer1.SingSong1" to the "Christmas" event of the Calendar object.

The result of the programming: "Singer1" starts to sing "Silent Night" at Christmas.

The metaphor:

WinFormApp1 project	Christmas Project
Form1	Singer1
Button1	Calendar
Close method of Form1	Sing Song method of Singer1
Action Form1.Close1 to close Form1	Action Singer1.SingSong1 to sing Silent Night
Click event of Button1	Christmas event of Calendar
Assign action Form1.Close1 to event Click of Button1	Assign action Singer1.SingSong1 to event Christmas of Calendar. That is, let Signer1 sing Silent Night at Christmas.

One difference in this metaphor is that the action Form1.Close1 does not use parameters while action Singer1.SingSong1 uses a parameter "Silent Night".

If an action requires parameters then Limnor Studio will allow you to visually specify parameter values. See http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20Part%20II.pdf

Visual and Codeless Programming

We just did a simple programming. Yet it shows the core concepts and major operations of visual and codeless programming in Limnor Studio.

The programming we did

We created a Windows Application. Limnor Studio generated a project for us. The project consists of two objects. One object is named WinApp which represents a Windows Form application. Another object is named Form1 which represents a window, or called a Form.

We developed object Form1 by adding a button named Button1 to it.

We changed the Text property of Button1 to "Close".

Note that there is nothing special about the above programming. Most visual programming systems currently on the market are doing the same.

What makes "codeless programming" possible is the following unique operations invented in the Limnor codeless programming system.

We used "Close" method of object Form1 to create an action named Form1.Close1.

We assigned the action Form1.Close1 to the Click event of object Button1.

This is the core concept of codeless programming: create actions and assign actions to events. It forms the foundation for Limnor Studio.

The concepts we used

Object – Basic programming entity, like a Lego brick. It is defined by properties, methods, and events.

Property – A value associated with an object and representing one aspect or characteristic of the object.

Method – Represents one thing an object can do.

Event – A specific moment occurs to an object.

Action – It indicates some concrete and specific task to be actually carried out.

Assign actions to events – An action will be executed only when it is assigned to an event and the event occurs.

Software Libraries

Limnor Studio is a tool for assembling objects to create new software. The objects are in software libraries.

The key point here is that if you are looking for specific computer functionality, such as showing a text or image on screen, getting data from a serial port, getting data from database, playing a video, capturing webcam video, etc., then you need to search in software libraries for appropriate objects. If you are looking for the ways to assembly objects together to make the objects behave as you want then you need to check Limnor Studio documentations to learn the appropriate operations.

 It is not possible for any one software developer to know all software libraries. There are large numbers of software libraries available. There are new software libraries keep coming from companies and individuals.

You do not need to know all software libraries. You just need to know the software libraries needed for your projects.

On the other hand the number of operations to assembly the software objects in software libraries are limited. Limnor Studio Users' Guide uses many examples to explain these operations. You will notice that most of operations used in different samples are the same. Each sample may introduce one or two unique operations, and the rest of operations are the same as used in other samples.

So, you can quickly learn all the functionality provided by Limnor Studio. Using this limited functionality provided by Limnor Studio, you can explore and exploit the unlimited functionality provided by software libraries.

In the sample above, we used the following operations:

- ➢ Create a project
- ➢ Modify object property
- ➢ Create action
- ➢ Assign action to event

You will repeatedly use the above operations again and again when developing your software.

Visual Programming in Limnor Studio

Visual programming languages may not necessarily be easy to learn. Some visual programming languages presented in research papers are hard to master even for a professional.

Another problem for a visual programming approach is its limitation of suitability. Unlike a textural programming language, which can be used for any programming tasks, a visual programming approach is only suitable for certain tasks.

Limnor Studio solves the problems by being a host of visual programming approaches. It hosts **independent** visual programming approaches. Any visual programming approach can be removed without affecting the operations of remaining visual programming approaches. Adding new visual programming approaches to Limnor Studio also does not affect the existing visual programming approaches. Limnor Studio coordinates all the visual programming approaches by sending signals notifying the programming changes the user makes via any visual programming approach. Each visual programming approach should handle the signals appropriately.

Different visual programming ideas can be implemented as visual programming approaches and insert into Limnor Studio. Thus it solves the problem of limitation of suitability if only one visual programming approach is used.

Some visual programming approaches shipped with Limnor Studio are listed below.

- **User Interface Designer** – It allows you to develop graphic user interface visually. You may add and remove objects. You may also create actions and assign actions to events. But you cannot remove actions or remove actions from events.
- **Object Explorer** – It shows all objects in hierarchy. You may create properties, methods and events. You may also create actions and assign actions to events. You may remove actions from events. You may delete actions. But you cannot add or remove objects.
- **Event Path** – It shows a control-flow diagram by showing the event and action paths. You may create properties, methods and events. You may also create actions and assign actions to events.

You may remove actions from events. You may delete actions. But you cannot add or remove objects.

More visual programming approaches are coming.

Further Readings

Further readings provide more information on how to use Limnor Studio.

Supported Project Type

This document explains what kinds of software projects are currently supported.

http://www.limnor.com/support/ProjectTypes.pdf

Basic Skills

The following documents explain the basic programming skills using Limnor Studio.

Basic programming concepts	Limnor Studio - User Guide - Part I.pdf
Use Action Parameters	Limnor Studio - User Guide - Part II.pdf
Use Expressions	Limnor Studio - User Guide - Part III.pdf
Create Properties, Methods, and Events	Limnor Studio - User Guide - Part IV.pdf
Use Method Editor	Limnor Studio - User Guide - Part V.pdf
Use Databases, data-binding, data entry, transaction, reporting	Limnor Studio - User Guide - Part VI.pdf
Use 2D drawings	Limnor Studio - User Guide - Part VII.pdf
Development of Windows Form applications	UseDefaultFormInstance.pdf
Handle events for large numbers of objects	HandleEventsForManyObjects.pdf
Resource Manager. Multiple Languages. Globalization and localization	Limnor Studio - User Guide - Resource Manager.pdf
Make Installer. Use a Setup project to generate msi file for software product installation	Limnor Studio - User Guide - Installer.pdf
Use file paths in actions, file paths are automatically adjusted on installing to users' computers	Limnor Studio - Non-Embedded Files.pdf

Samples of using libraries

Use 3rd party libraries. Create constructors.	UseDb4o.pdf
Use external libraries with generic types and native libraries	UseGenericTypes.pdf
How to use native libraries	UseNativeLibraries.pdf
How to use libraries.	HowToLoadTypes.pdf
Work with callback functions and constructors (NDDE samples)	NDDE_samples.pdf
How to connect to Firebird database	UseFirebird.pdf

Create Installers

The following document explains how to create an installer for distributing your software.

http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20Installer.pdf

Add Copy-Protection

The following document explains how to add copy-protection to your Limnor Studio generated software and issuing licenses to your customers.

http://www.limnor.com/support/CopyProtection.pdf

Handle Exceptions

The following document explains how to handle exceptions in your software.

http://www.limnor.com/support/HandleExceptions.pdf

Create Screen Savers

The following document explains how to create and distribute screen savers.

http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20ScreenSaver.pdf

How to use TreeView

The following document explains how to use TreeView control in a Windows Form project.

http://www.limnor.com/support/TreeViewX.pdf

Note that the above document is not for web projects. To use TreeView in a web project, see

http://www.limnor.com/support/WebTreeView.pdf

Database Programming for standalone applications

Use Databases, data-binding, data entry, transaction, reporting	Limnor Studio - User Guide - Part VI.pdf
How to connect to MySQL using ADO.Net	ConnectToMySqlWithADO.pdf
How to connect to MySQL using ODBC	ConnectToMySqlWithODBC.pdf
Desktop Database Security	Desktop Database Security.pdf
How to do tiered database programming.	TieredDatabaseProgramming.pdf
How to bind image data to picture box.	DataBindingForImages.pdf
How to transfer data	dataTransfer.pdf
How to set background color of rows in EasyGrid	SetRowColor.pdf
How to locate row in an EasyGrid by criteria.	SearchInEasyGrid.pdf

Note that the above documents are for standalone applications. For web applications, see "Web Application Developments" section.

Web Application Developments

Development of web applications	WebApplicationDevelopment.pdf
Development of web database applications	webDatabaseProgramming.pdf
How to use data repeater in web applications	WebDataRepeater.pdf
How to use tree view in web applications	WebTreeView.pdf
How to use arbitrary PHP code/files in PHP web applications	UsePhpCode.pdf
How to use arbitrary JavaScript code/files in web applications	UseJavaScriptFiles.pdf
How to add file-uploaders in PHP web applications	WebFileupload.pdf
How to create web mailer in PHP web applications	WebEmail.pdf
How to use menus in web applications	WebMenu.pdf
How to use dialogue boxes and child windows in web applications	WebDialogAndChildPage.pdf
How to use date times in web pages	WebDateTime.pdf
How to bind images saved in database in BLOB's	WebImageDataBinding.pdf
How to make use of web event bubbling	http://www.limnor.com/support/WebEventBubbling.pdf
How to handle large number of web elements	http://www.limnor.com/support/HandleLargeNumberOfWebElements.pdf

Work with Microsoft Office

How to use Microsoft Office Interfaces - Word sample 1.	UseWord.pdf
How to use Microsoft Office Interfaces - Word sample 2.	UseWord2.pdf
How to use Microsoft Office Interfaces - Excel sample.	UseExcel.pdf

How to …

The "How to …" web page lists documents dealing with a wide range of specific topics.

See

http://www.limnor.com/studio_howto.html

Document center

Most of documents can be found at http://www.limnor.com/LimnorTutorial.html

Feedback

Please send your feedback to support@limnor.com