

Properties, Methods and Events

What are they ?

The buzzwords Properties, Methods and Events owe their origins largely to Object Oriented methodologies and technologies. Crimson Basic is designed to avoid the complexities of Object Oriented methods, however, the GUI controls used are implemented as objects and therefore these terms are used to describe the way in which the programmer interacts with GUI controls.

Properties

Properties are a way of storing information about a particular instance of a control, for example, a Button control may appear many times within a form, but each Button has distinct characteristics (properties) depending upon how the programmer wishes the button to be used. One Button on a form may be Visible whereas another may be Invisible. Therefore 'Visible' is a property of Button. Properties are effectively variables which exist in their own right for each control.

Properties are Set and retrieved using the following format :

Formname.Controlname.Property

Examples of the Setting and Getting of Properties are :

MyForm.Button1.Enabled=True

(Sets the Enabled Property on Control Button1 on Form MyForm to True. The button will now respond to mouse clicks etc.)

X=Myform.Button1.Visible

(X is set to the True/False value of the Visible Property of Button1 on Myform.)

MyForm.EText.Value="Hello"

(Sets the Value of the text in the Editable Text field named EText on the Form MyForm to "Hello")

Methods

A Method is very much like a subroutine except that it relates to a particular instance of a Control. Methods are used to perform operations on Controls which cannot be performed by simply setting Properties, such as :

Form1.OpenWindow()

(Opens the Window named Form1 [this is the name you gave it when you designed it] and displays it [if the Visible Property is set to True])

Form1.ShowWindow()

(Makes the Form Form1 visible on the screen if it was invisible. In this case, calling the ShowWindow method is equivalent to setting the Forms Visible Property to True).

X=MyForm.MyListBox.GetRow(2)

(Sets the String X to the value stored in the second row within the ListBox MyListBox on the Form MyForm.

MyForm.MyListBox.InsertRow(2,"Pot Noodle")

(Inserts a row in the ListBox MyListBox, after row 2, containing the string "Pot Noodle").

Events

An Event is something which occurs due to a user of the program using the mouse or pressing a key in some way. These Events are reported to the program as calls to Procedures which are coded by the programmer. For example, If the Form Form1 contains a Button named Button1 the following procedure will be called every time the user Clicks on that particular button.

```
Procedure Form1.Button1.Click()  
.....  
.....  
Return
```

Other user events may be processed in the same way :

```
Procedure Form1.OnMenu(MenuId,MenuItem)  
Parameter MenuId As Integer  
Parameter MenuItem As Integer  
.....  
.....  
Return
```

(This procedure will be called every time the user selects a menu item whilst this form is active [the one in front]. The MenuId and MenuItem relate to the numbers which were supplied when the menus were created).

```
Procedure MyForm.Rect1.MouseDown(XPos,YPos)  
Parameter XPos As Integer  
Parameter YPos As Integer  
.....
```

.....
Return

(This procedure will be called every time the user depresses the mouse whilst it is within the Rectangle named Rect1 on MyForm. The Xpos and YPos parameters contain the Horizontal and Vertical position of the mouse (in pixels) within the Rectangle at the time the mouse button was pressed.