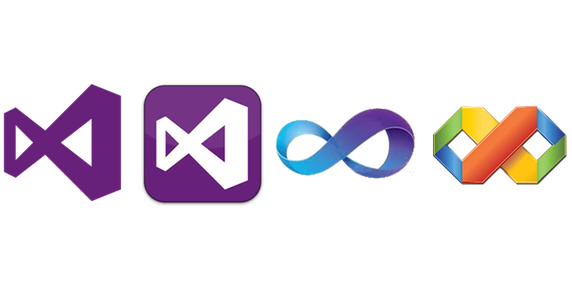


**Characteristics of event driven programming**

To: Micheal Kimber

From: Callum Pope

Date:26/09/16

Table of Contents

[Introduction 3](#_Toc463013342)

[Characteristics of Visual Basic 3](#_Toc463013343)

[Features available on Visual Basic 5](#_Toc463013344)

[Advantages of Visual Basic for the development of event driven programs 7](#_Toc463013345)

[Bibliography of sources 8](#_Toc463013346)

# Introduction

“Event driven programming is a computer programming paradigm in which control flow of the program is determined by the occurrence of events” (What Is Event-Driven Programming?, 2016)

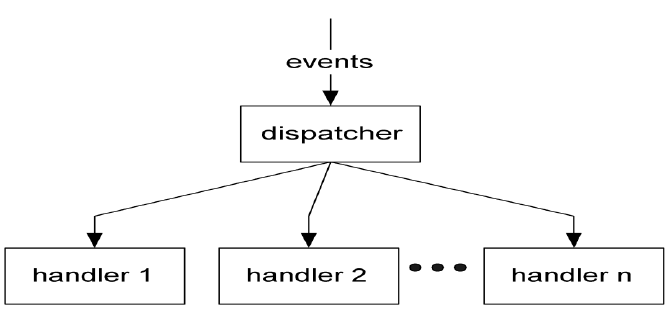
This guide will cover the features of the event driven programming environment Microsoft Visual Basic explaining and detailing all available features and tools that can be used to aid the process of development. All advantages of using this will also be explained.

# Characteristics of Visual Basic

A typical event driven programming language has 4 main characteristics. These are:

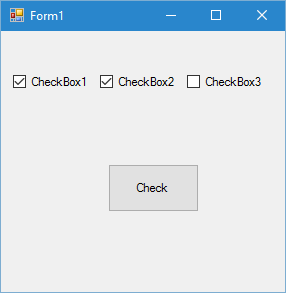
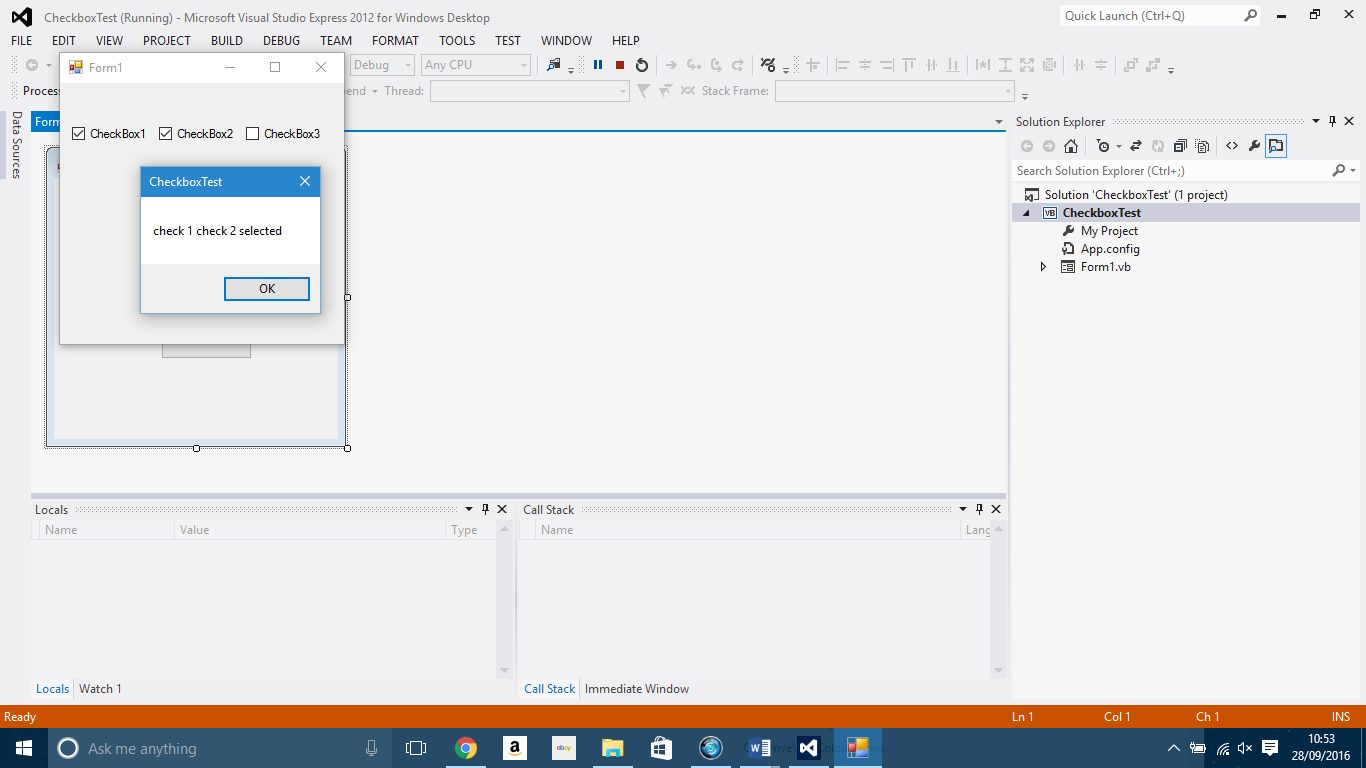
* Event loops – continually waits for an event to happen
* Triggers –” used by event loops to identify and launch a response to an event which has happened” (Kimber, 2016)
* Forms – integrated environment where components can be placed (buttons, text, labels)
* Event handlers – code designed to deal with an event

I will be discussing each of these in detail and how they relate to visual basic.

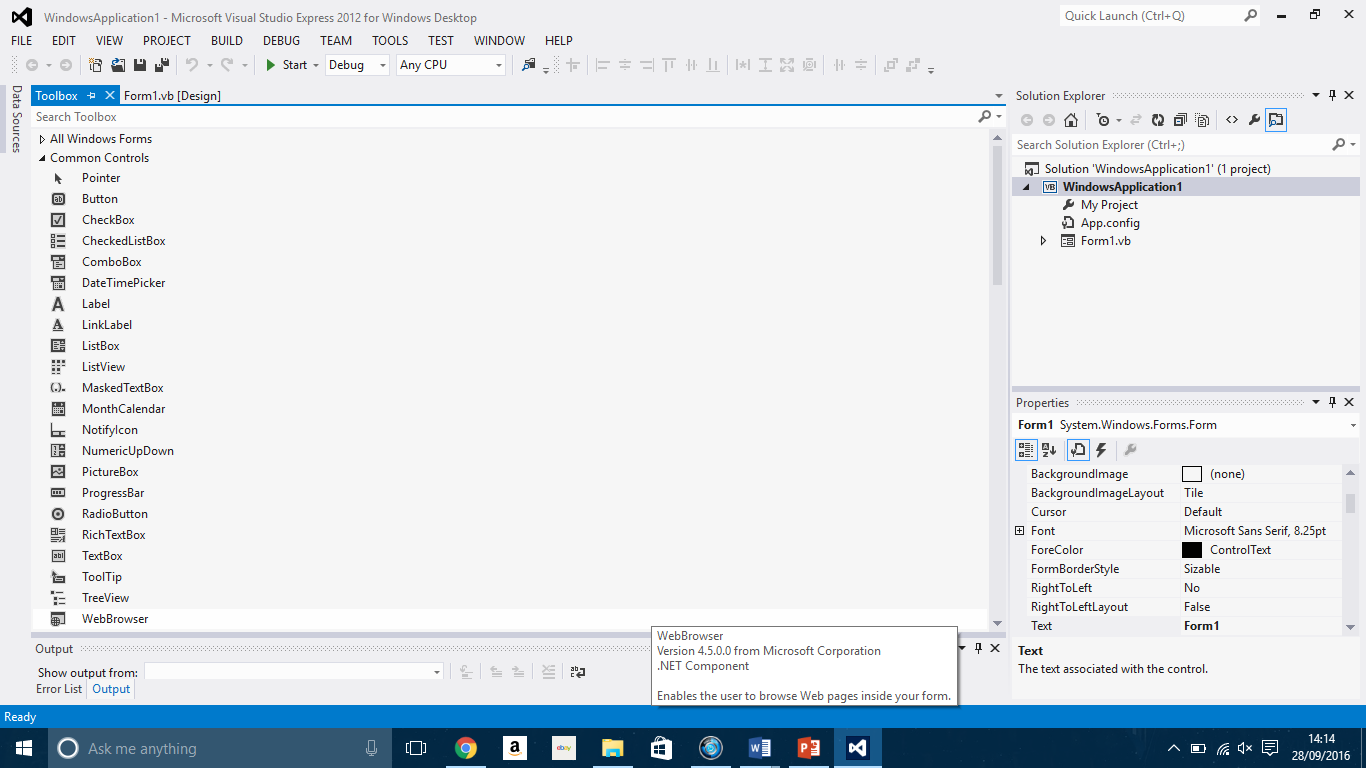
An event loop is a continuous cycle in which events are constantly being searched for. An event loop typically consists of 3 sections: events, a dispatcher and multiple event handlers. An example can be seen in the diagram below.

An event loop is used in tandem with triggers that identify an event made by either the user or the system where a response is required. “Events can be sub categorized into five different categories which refers to different hardware and software uses of events. These events are: mouse events, keyboard events, HTML object events, Form events and User Interface Events.” (Burrows, 2015). An example of each of these could be: mouse (clicking a button), keyboard (Ctrl, Alt, delete to log out of system), HTML (onclick), form (button press) and user interface events. This triggered event is then picked up by the dispatcher that decides what event handler, or section of code, would be best suited to carry out the created event. All unprocessed events that are waiting for an event handler to become free are stored in an event queue. “An event queue is a repository where events from an application are held prior to being processed by a receiving program or system” (Divestopedia, 2016). This event is then dispatched to the appropriate handler that can execute the task. An event hander is physical code that has been produced that can execute the event and launch a response to it.

For example, when a button on a form is clicked, then an event is triggered that requires a response from the system. In the scenario below the form allows the user to select checkboxes. Once the check button has been pressed an event is triggered that requires a system response.

When this button has been pressed and recognized by the dispatcher, it will then have to be checked in order for the event to be delivered to the appropriate event handler selected by the dispatcher. In this case the event would be sent to a handler best suited to the check button.” Event handlers are procedures that are called when a corresponding event occurs. You can use any valid subroutine with a matching signature as an event handler. You cannot use a function as an event handler, however, because it cannot return a value to the event source” (Microsoft, 2016a). Once the handler has been selected, it will execute the code that the user has written and return a result to the user on screen.

This screenshot shows the response from the event handler to the selection of the check button. The code created states that there must be a message displayed back on screen to the user in the form of a message box. This message box should show the check boxes that have been checked only.



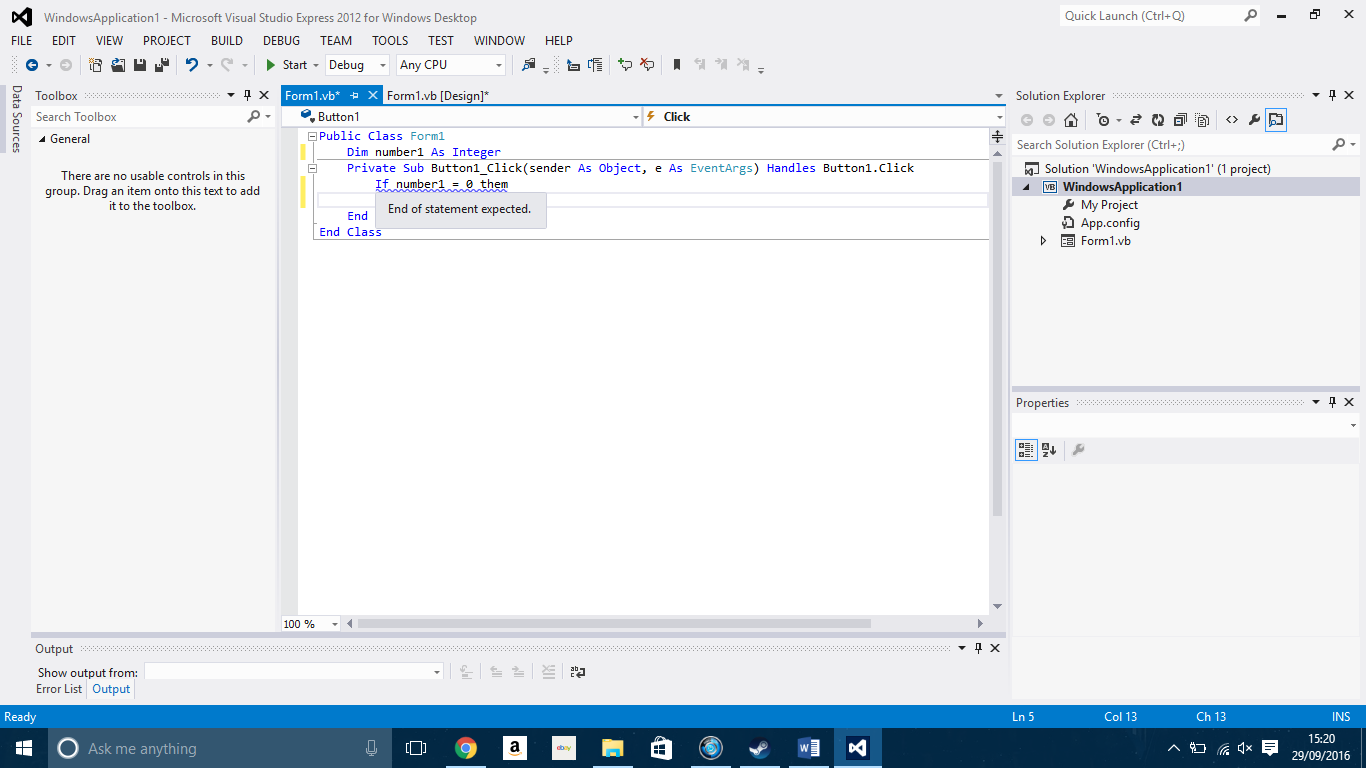
A form as shown in the example above is an integrated development environment where the user can place object controls or buttons on screen. A typical form could include many components such as labels, buttons, text boxes etc. Some of these controls are unique to visual basic and will be discussed in further detail.

These controls work hand in hand with the forms on visual basic as no one can work without the other. These controls can be found within the toolbox that allows the user to place objects and interact with the form.

# Features available on Visual Basic

Alongside the basic features of event driven programming languages previously discussed, there are also features that are unique to the visual basic environment that include the controls discussed previously that can aid the development process.

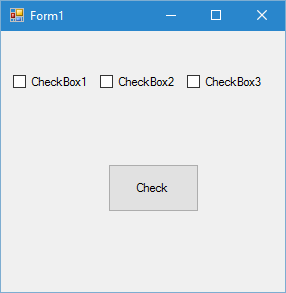
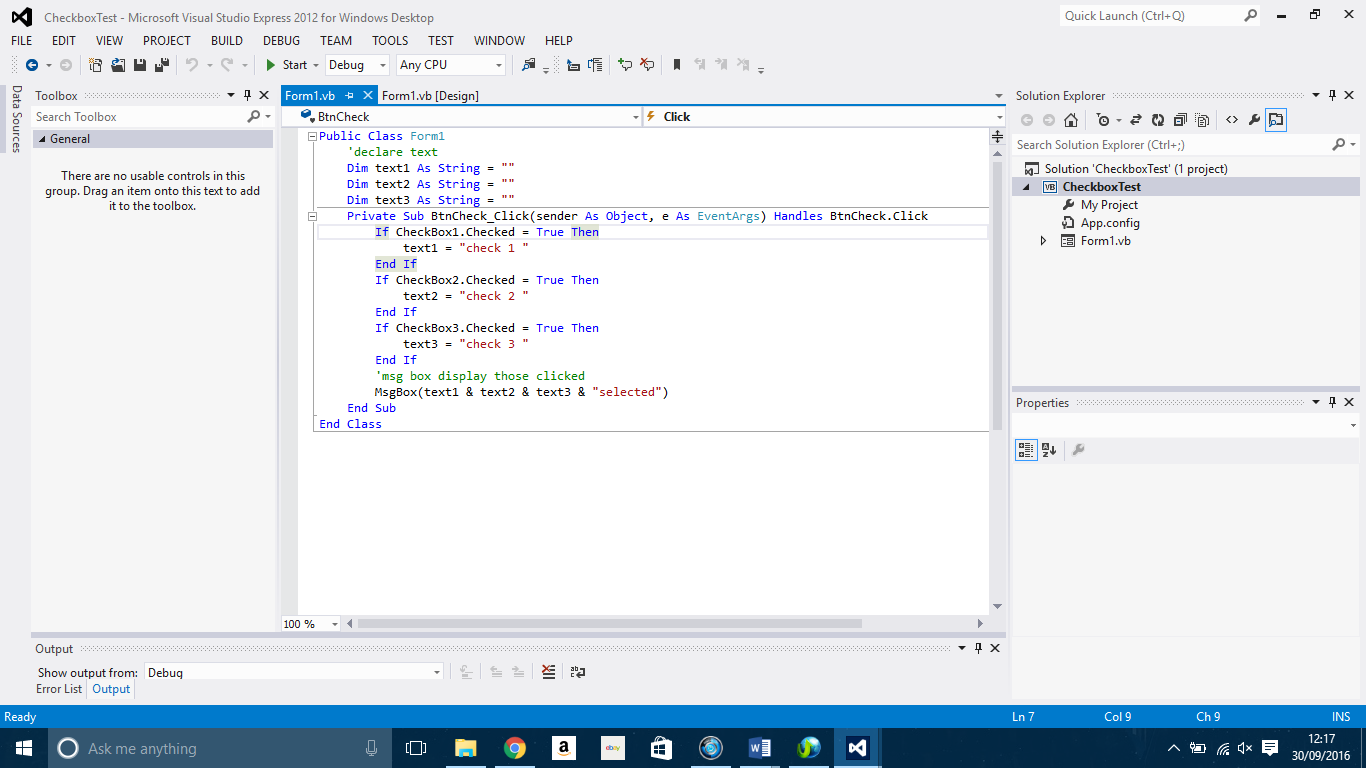
One of these features that are available is autocomplete. This gives the user suggestions on words that can be used from what they have already written. These can either be known words like for or if or can be user declared names such number1. This is because the names are stored making writing the program much faster and easier as the user does not have to type out the entire word once the suggestion has appeared. As a result, this can make syntactical error checking much easier as errors can be spotted much easier as they are underlined in blue.



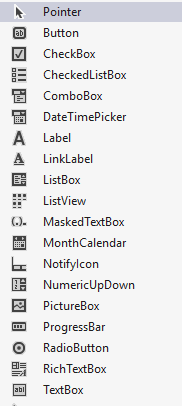
In the example above the known word “then” has been spelt incorrectly, causing the blue underline to appear highlighting that it is an error. The program will be unable to run until this error has been corrected.

Another feature of visual basic is the debugging feature. This allows you to run your program as you are creating it, testing out the various sections of code created for individual buttons or forms. If there are any errors within the code, then debugging will stop and errors will be displayed in an error list at the bottom of the screen. These errors will have a description of what they are and a possible solution to the error. It is then up to the user to correct these errors and rerun the program to see if they have been corrected. Debugging code also allows for the user of break points that allows you to execute code up until a certain point or until an error is reached within that section of code. The debugging feature can be accessed here. (Microsoft, 2016b)



The code window itself is a feature of visual basic. This code window splits the code into subroutines or procedures that each deal with a particular task. For example, a button named check would contain code on what would happen when that button is clicked.

Pictured above are both the form including the check button and its corresponding code.

The toolbox feature allows the user to add to a form. There are many basic tools that can be used in visual basic on a form:

Button – adds a control button to the form that can be renamed and programmed to perform a specific task once clicked.

CheckBox – allows the user to check off from a list of options or toggle a setting on or off.

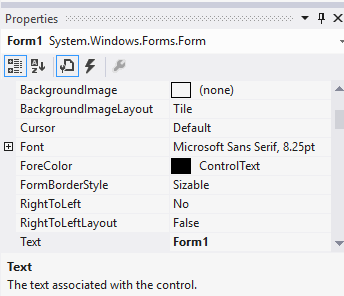
Label – used for adding text that cannot be edited to the form for possibly a title or field name.

PictureBox – allows the user to add a picture to the form.

RadioButton – can be used for the same function to toggle a setting on or off but are mainly used for multiple choice options.

TextBox – allows the user to type on the form. This can be used in scenarios such as entering data to a database or just adding numbers together.

There is also a properties window that allows you to change the properties window that allows you to change the properties of both the form and any objects placed on it. You can change the names of objects on the form, the text that is displayed on them, the colour of objects and the form, the size of objects such as labels or text boxes, the visibility of an object or whether it is able to be edited.



# Advantages of Visual Basic for the development of event driven programs

There are many advantages of using an IDE such as visual basic for the development. These include but are not limited to:

* Being able to design a program suitable to a user interface, using objects such as text boxes or buttons.
* Makes use of already created code (library routines) for searching and sorting.
* The tools and utilities available help with rapid application development (RAD) of programs

# Bibliography

Burrows, R. (2015) Key features of event driven programming. Available at: <http://reece-burrows-media.blogspot.co.uk/2013/11/key-features-of-event-driven-programming.html> (Accessed: 30 September 2016).

Butler, D. and Mullins, D. (1999) CeX (UK): About CeX. Available at: <https://uk.webuy.com/about/history.php> (Accessed: 3 October 2016).

Divestopedia (2016) What is an event queue? - definition from Techopedia. Available at: <https://www.techopedia.com/definition/24963/event-queue> (Accessed: 30 September 2016).

Kimber. M., 2016. “Unit 20: Event-driven Programming Features and Characteristics”. PowerPoint Presentation. City of Liverpool College. 26 Sept 2016.

Microsoft (2016a) Events and event handlers. Available at: <https://msdn.microsoft.com/en-us/library/2z7x8ys3(v=vs.90).aspx> (Accessed: 2016).

Microsoft (2016b) Debugging your visual basic application. Available at: <https://msdn.microsoft.com/en-us/library/ms172744(v=vs.110).aspx> (Accessed: 30 September 2016).

What is event-driven programming? (2016) Available at: <http://www.computerhope.com/jargon/e/event-driven-prog.htm> (Accessed: 30 September 2016).