

Events are user actions such as key press, clicks, mouse movements, etc., or some occurrence such as system generated notifications. Applications need to respond to events when they occur. For example, interrupts. Events are used for inter-process communication.

Using Delegates with Events

The events are declared and raised in a class and associated with the event handlers using delegates within the same class or some other class. The class containing the event is used to publish the event. This is called the **publisher** class. Some other class that accepts this event is called the **subscriber** class. Events use the **publisher-subscriber** model.

A **publisher** is an object that contains the definition of the event and the delegate. The event-delegate association is also defined in this object. A publisher class object invokes the event and it is notified to other objects.

A **subscriber** is an object that accepts the event and provides an event handler. The delegate in the publisher class invokes the method (event handler) of the subscriber class.

Declaring Events

To declare an event inside a class, first of all, you must declare a delegate type for the event as:

```
public delegate string BoilerLogHandler(string str);
```

then, declare the event using the **event** keyword –

```
event BoilerLogHandler BoilerEventLog;
```

The preceding code defines a delegate named *BoilerLogHandler* and an event named *BoilerEventLog*, which invokes the delegate when it is raised.

Example

```
using System;

namespace SampleApp {
    public delegate string MyDel(string str);

    class EventProgram {
        event MyDel MyEvent;

        public EventProgram() {
            this.MyEvent += new MyDel(this.WelcomeUser);
        }
        public string WelcomeUser(string username) {
            return "Welcome " + username;
        }
    }
}
```

```
static void Main(string[] args) {  
    EventProgram obj1 = new EventProgram();  
    string result = obj1.MyEvent("Tutorials Point");  
    Console.WriteLine(result);  
}  
}
```

When the above code is compiled and executed, it produces the following result –

Welcome Tutorials Point

Previous Page **Print Page**

Next Page

Advertisements