

LESSON 1.2

10754 Microsoft .NET Fundamentals

Understand Events and Event Handling in the .NET Framework

Lesson Overview

How are events used in .NET applications?

In this lesson, you will explore:

- Event-driven programming
- Raising and handling events
- Delegates

Guiding Questions

1. What is event-driven programming?
2. How are events raised and handled in .NET applications?
3. What are delegates?

Anticipatory Set

It is likely that your day began with an event: an alarm clock going off. You responded to that event by getting out of bed—or perhaps by pressing “snooze,” causing the event to repeat!

What are some other events that happen in our daily lives, and how do we respond to those events?

Event-driven Programming

- Event-driven programming is a model or paradigm in which the flow of program execution is determined by events.
 - Often, these events are forms of user input (such as mouse clicks, key presses, or touch-screen interactions).
- An **event** is like a signal to another part of the application.
- The developer can create **event handlers** that will respond to those signals.
- A simple example is a Save button on a Microsoft® Windows® form.
 - When the user clicks the button, that causes a “signal” to be sent to code that saves the current data.
 - In this example, the button click raises (or sends) an event; an event handler receives the event and saves the data.

Events

- Events are actions that occur in a program that the developer can respond to, or handle, in code.
- When an event occurs, we say that it has been “raised.”
- In addition to mouse clicks, what are some other events in .NET applications?

Event Handlers

- Event handlers are methods or procedures that are invoked when the corresponding event is raised.
- In addition to the terms “raised” and “handled,” developers may also use the terms “publish” and “subscribe.”
 - A class can publish an event, and another class can subscribe to that event.

Creating an Event Handler in the Designer

1. Click the form or control—such as a Button—that you want to create an event handler for.
2. In the Properties window, click the Events button (the lightning-bolt icon).
3. In the list of available events, click the event that you want to handle. For example, for a Button, you might select the Click event.
4. In the box to the right of the event name, type a name for the handler and press ENTER.
5. The Designer will create the procedure or method and will register it as an event handler. Add the appropriate code to this new code block.

Note: In Microsoft Visual Basic®, you can select the object and event from the drop-down lists at the top of the Code Editor; this will create a blank event handler for the selected event.

Delegates

- Delegates are objects that refer to methods.
- In political diplomacy, a delegate is a person authorized to represent another (a person or an organization), such as when a government sends delegates to represent the country at the Olympics. Likewise, a delegate in this case stands in for, or represents, a method.
- .NET applications use delegates to connect events to event handlers.
 - The event uses a delegate that refers to the method that handles the event.
 - A strength of the delegate system is that it allows developers to link events to handlers *dynamically*; that is, an event can be handled by different methods depending on the current situation.

LESSON 1.2

10754 Microsoft .NET Fundamentals

Assignment

Create the form described on the Student Activity document, including the described event handlers.

Lesson Review

Explain the event-driven programming model, including the roles of events and event handlers.