Event Handlers Overview (Windows Forms)

.NET Framework 4.5 3 out of 5 rated this helpful

An event handler is a method that is bound to an event. When the event is raised, the code within the event handler is executed. Each event handler provides two parameters that allow you to handle the event properly. The following example shows an event handler for a Button control's Click event.

C#

```
private void button1_Click(object sender, System.EventArgs e)
{

}
```

The first parameter,sender, provides a reference to the object that raised the event. The second parameter, e, in the example above, passes an object specific to the event that is being handled. By referencing the object's properties (and, sometimes, its methods), you can obtain information such as the location of the mouse for mouse events or data being transferred in drag-and-drop events.

Typically each event produces an event handler with a different event-object type for the second parameter. Some event handlers, such as those for the MouseDown and MouseUp events, have the same object type for their second parameter. For these types of events, you can use the same event handler to handle both events.

You can also use the same event handler to handle the same event for different controls. For example, if you have a group of RadioButton controls on a form, you could create a single event handler for the Click event and have each control's Click event bound to the single event handler. For more information, see How to: Connect Multiple Events to a Single Event Handler in Windows Forms.

See Also

Concepts

Events Overview (Windows Forms)

Other Resources

Creating Event Handlers in Windows Forms

Was this page helpful? ○ Yes ○ No