

Using Delegates and Events



Jesse Liberty

@jesseliberty <http://jesseliberty.com>



A delegate is a reference
to a method



This can be confusing,
hang in there.



```
public class MediaInventory {  
    public delegate bool TestMedia();
```

Step 1 TestMedia

Declare the delegate

```
public void TestResult(TestMedia mediaDelegate) {  
    if (mediaDelegate() == true) {  
        Console.WriteLine("Media works fine, add to inventory");  
    } else {  
        Console.WriteLine("Unable to play media. Rejected!");  
    }  
}
```

Step 2 MediaDelegate

Create a Method to Use the Delegate

```
public bool PlayRecord() {  
    Console.WriteLine("Testing the record. Works fine");  
    status = true;  
    return status;  
}
```

Step 3 RecordPlayer.cs

Create One or More Methods That Match the Delegate's Return Value and Parameters

```
public bool TestCassette() {  
    Console.WriteLine("Testing Cassette... Failed!");  
    return false;  
}
```

Step 3a CassettePlayer.cs

Create Another Method That Matches the Delegate's Return Value and Parameters

```
TestMedia testRecordDelegate =  
    new TestMedia(recordPlayer.PlayRecord);
```

Step 4 Worker.cs

Instantiate the Delegate


```
TestResult(testRecordDelegate);
```

Step 5 Worker.cs

Invoke the Method Through the Delegate

Demo



Delegates



An event tells you that
something happened



Creating and Subscribing to a Clock

The clock

A digital clock

TimeInfoEventArgs

The client



Custom EventArgs

```
public class TimeInfoEventArgs {  
    public int Hour { get; set; }  
  
    public int Minute { get; set; }  
  
    public int Second { get; set; }  
}
```



Declaring an Event

```
public class Clock {  
  
    public delegate void SecondChangeHandler(object clock,  
        TimeInfoEventArgs timeInformation);  
  
    public event SecondChangeHandler SecondChanged;  
}
```

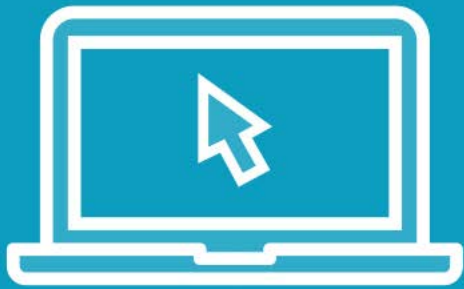


Subscribing to the Event

```
public class DigitalClock {  
    public void Subscribe(Clock theClock) {  
        theClock.SecondChanged += NewTime;  
    }  
}
```



Demo



Events



Module Summary



Delegates

Events

Subscribers

