

```
1 using System;
2 using System.Threading;
3
4
5 namespace My_Threading
6 {
7     public class Thread2
8     {
9         public int _start = 0;
10        public void Tread2()
11        {
12            string module = "Thread2->Konstructor: ";
13            Console.WriteLine("{0}", module);
14
15        }
16        public int Start
17        {
18            get { return _cmd; }
19            set { _cmd = value; }
20        }
21
22        public object name = "Thread2";
23        private int _cmd = 0;
24        public delegate void myDelegat(int t);
25        public static event myDelegat TestEvent;
26
27        public delegate void textDelegat(string t);
28        public static event textDelegat TextEvent;
29        public delegate void EventHandler3<MoreEventArgs>(object sender, MoreEventArgs e);
30        public static event EventHandler3<MoreEventArgs> Msg_Subscriber_Thread3;
31
32
33
34        public void RaiseTheEvent2()
35        {
36            string module = "Thread2->RaiseTheEvent2: ";
37            while (true)
38            {
39                Thread.Sleep(500);
40                if (_cmd > 0)
41                {
42                    Console.WriteLine("{0}", module);
43                    //TestEvent?.Invoke(_cmd);
44                    TextEvent?.Invoke("Text from Thread2 Cmd = " + _cmd.ToString());
45                    // Using eventhandler with more arguments
46                    MoreEventArgs args = new MoreEventArgs
47                    {
48                        command = _cmd,
49                        msg = "Message with MoreEventArgs()"
50                    };
51                }
52            }
53        }
54    }
55 }
```

```
51
52         Msg_Subscriber_Thread3?.Invoke(name, args);
53         _cmd = 0;
54     }
55 }
56
57 }
58
59 public void Subscriber2(int m)
60 {
61     _cmd = m;
62     string module = "Thread2->Subscriber2: ";
63     Console.WriteLine("{0}Event from Th1 was rised with {1}",    ↗
        module, m);
64 }
65 }
66 }
67
```