

# Example21~23

## CCR Failure Handling

### Example 21: Causalities

```
1
2
3
4 using System;
5 using System.Collections.Generic;
6 using System.Text;
7 using Microsoft.Ccr.Core;
8
9 namespace Example
10 {
11     class Program
12     {
13         // create dispatcher and dispatcher queue for scheduling tasks
14         static Dispatcher dispatcher = new Dispatcher();
15         static DispatcherQueue _taskQueue = new DispatcherQueue("sample queue", dispatcher);
16
17         // Example 21: Causalities
18         static void Main(string[] args)
19         {
20             /// <summary>
21             /// 平行運作中發生的錯誤，將丟出給錯誤處理
22             /// </summary>
23             Port<Exception> exceptionPort = new Port<Exception>();
24
25             // create a causality using the port instance
26             Causality exampleCausality = new Causality("root cause", exceptionPort);
27
28             // add causality to current thread
29             Dispatcher.AddCausality(exampleCausality);
30
31             // any unhandled exception from this point on, in this method or
32             // any delegate that executes due to messages from this method,
33             // will be posted on exceptionPort.
34
35             Port<int> portInt = new Port<int>();
36             Arbiter.Activate(_taskQueue,
37                 Arbiter.Receive(true, portInt, IntHandler)
38             );
39
40             // causalities flow when items are posted or Tasks are scheduled
41             portInt.Post(1);
42             portInt.Post(0);
43
```

```

44         // activate a handler on the exceptionPort
45         // This is the failure handler for the causality
46         Arbiter.Activate(_taskQueue,
47             Arbiter.Receive(true, exceptionPort,
48                 delegate(Exception ex)
49                     {
50                         // deal with failure here
51                         Console.WriteLine(ex);
52                     })
53             );
54     }
55
56     static void IntHandler(int i)
57     {
58         // print active causalities
59         foreach (Causality c in Dispatcher.ActiveCausalities)
60         {
61             Console.WriteLine(c.Name);
62         }
63
64         // expect DivideByZeroException that CCR will redirect to the causality
65         Console.WriteLine("i: " + i);
66         int k = 10 / i;
67     }
68 }
69 }

```

## 70 Example 22: Nested Causalities

```

71 using System;
72 using System.Collections.Generic;
73 using System.Text;
74 using Microsoft.Ccr.Core;
75
76 namespace Example
77 {
78     class Program
79     {
80         //Example 15: Task.Execute
81         static void Main(string[] args)
82         {
83             /// <summary>
84             /// 與Example 16相同，但不透過port傳遞資料
85             /// 而是透過Task.Execute執行任務，直接執行任務
86             /// 當資料一傳入後，可馬上執行
87             /// </summary>
88
89             var dispatcher = new Dispatcher(
90                 0, // zero means use one thread per CPU, or 2 if only one CPU present

```

```

91         "sample dispatcher" // friendly name assigned to OS threads
92     );
93
94     var taskQueue = new DispatcherQueue(
95         "sample queue", // friendly name
96         dispatcher // dispatcher instance
97     );
98
99     // directly enqueue a task with an inlined method plus a parameter
100    taskQueue.Enqueue(
101        new Task<int>(5, item => Console.WriteLine(item))
102    );
103    }
104 }
105 }

```

## 106 Example 23: Joins and Causalities

```

107 using System;
108 using System.Collections.Generic;
109 using System.Text;
110 using Microsoft.Ccr.Core;
111
112 namespace Example
113 {
114     class Program
115     {
116         // create dispatcher and dispatcher queue for scheduling tasks
117         static Dispatcher dispatcher = new Dispatcher();
118         static DispatcherQueue _taskQueue = new DispatcherQueue("sample queue", dispatcher);
119
120         //Example 23: Joins and Causalities
121         static void Main(string[] args)
122         {
123             PortSet<int, string> intPort = new PortSet<int, string>();
124             //Port<int> intPort = new Port<int>();
125             Port<int> leftPort = new Port<int>();
126             Port<string> rightPort = new Port<string>();
127
128             Port<Exception> leftExceptionPort = new Port<Exception>();
129             Port<Exception> rightExceptionPort = new Port<Exception>();
130
131             intPort.Post(1);
132             intPort.Post("2");
133             //intPort.Post(2);
134
135             // activate two handlers that will execute concurrently and create
136             // two different parallel causalities
137             Arbiter.Activate(_taskQueue,

```

```

138     Arbiter.Receive<int>(true, intPort,
139     delegate(int i)
140     {
141         Causality leftCausality = new Causality("left", leftExceptionPort);
142         Dispatcher.AddCausality(leftCausality);
143         Console.WriteLine("Left !");
144         leftPort.Post(i);
145     })
146 );
147
148 Arbiter.Activate(_taskQueue,
149     Arbiter.Receive<string>(true, intPort,
150     delegate(string i)
151     //Arbiter.Receive<int>(true, intPort,
152     //delegate(int i)
153     {
154         Causality rightCausality = new Causality("right", rightExceptionPort);
155         Dispatcher.AddCausality(rightCausality);
156         Console.WriteLine("Right !");
157         // post item on rightPort under the context of the right causality
158         rightPort.Post(i.ToString());
159     })
160 );
161
162 // activate one join receiver that executes when items are available on
163 // both leftPort and rightPort
164
165 Arbiter.Activate(_taskQueue,
166     Arbiter.JoinedReceive<int, string>(false, leftPort, rightPort,
167     delegate(int i, string s)
168     {
169         throw new InvalidOperationException("This exception will propagate to two peer
170 causalities");
171     })
172 );
173
174 // activate a handler on the exceptionPort
175 // This is the failure handler for the causality
176 Arbiter.Activate(_taskQueue,
177     Arbiter.Receive(true, leftExceptionPort,
178     delegate(Exception ex)
179     {
180         // deal with failure here
181         Console.WriteLine("Left causality: " + ex);
182     })
183 );
184
185 // activate a handler on the exceptionPort
186 // This is the failure handler for the causality
187 Arbiter.Activate(_taskQueue,
188     Arbiter.Receive(true, rightExceptionPort,
189     delegate(Exception ex)

```

```
190         {
191             // deal with failure here
192             Console.WriteLine("Right causality: " + ex);
193         })
194     );
195 }
196 }
197 }
```