

Asynchronous Programming with TPL: Overview

Ian Griffiths

<http://www.interact-sw.co.uk/iangblog/>

ian@interact-sw.co.uk



The Task Parallel Library (TPL)

Not only

Multithreading

but also

Asynchronous Programming



Asynchronous Defined

- **Concurrent**
 - Several things happening at once
- **Multithreaded**
 - Multiple execution contexts
- **Parallel**
 - Multiple simultaneous computations
- **Asynchronous**
 - Not having to wait



An Asynchronous Example



The Task Class

```
public class Stream
{
    ...
    public Task WriteAsync(byte[] buffer, int offset, int count)
    public void Write(byte[] buffer, int offset, int count)
    ...
    public int Read(byte[] buffer, int offset, int count)
    public Task<int> ReadAsync(byte[] buffer, int offset, int count)
    ...
}
```



Task vs Older Async Patterns

- **Easier to use than APM**

- Uniformity through generics
- Result property
- Continuations

.NET >= 1.0

APM: Asynchronous
Programming Model

.NET >= 2.0

EAP: Event-based
Async Pattern

- **Schedulers**

.NET >= 4.0

Tasks

- **Composition**

- **Error handling**

- **Cancellation**

- **Language support**



Error Handling

- **Properties**

- IsFaulted
- Status
- Exception
- Result

- **Wait methods**

- **AggregateException**

- **Unobserved exceptions**

- TaskScheduler.UnobservedTaskException
- Escalate

```
<configuration>  
  <runtime>  
    <ThrowUnobservedTaskExceptions  
      enabled="true"/>  
  </runtime>  
</configuration>
```



Continuations and Errors

- **TaskContinuationOptions**

- OnlyOnFaulted
- OnlyOnRanToCompletion
- OnlyOnCanceled
- NotOnFaulted
- NotOnRanToCompletion
- NotOnCanceled

- **Non error related:**

- PreferFairness
- LongRunning
- ExecuteSynchronously
- AttachedToParent

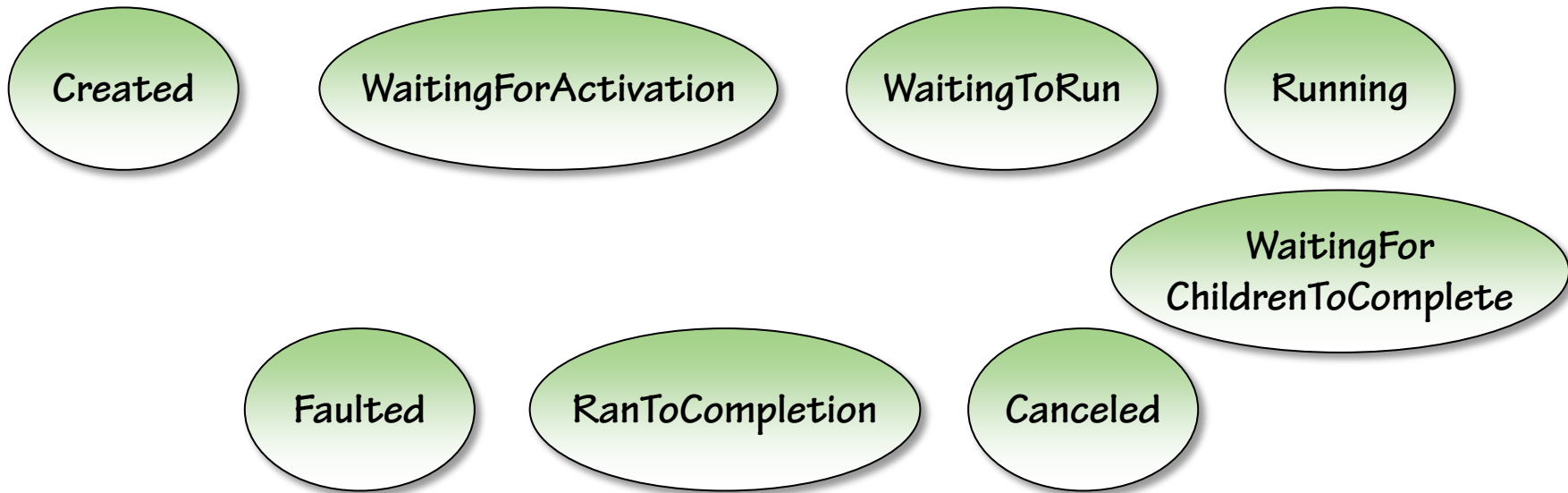


Composition

- **Task.WaitAll and Task.WaitAny**
- **TaskFactory.ContinueWhenAll and TaskFactory.ContinueWhenAny**
- **Task.WhenAll and Task.WhenAny**
 - Requires .NET ≥ 4.5



Task Status



Status	IsCompleted	IsFaulted	IsCanceled
WaitingForActivation, WaitingToRun, Created, Running, WaitingForChildrenToComplete	False	False	False
RanToCompletion	True	False	False
Faulted	True	True	False
Canceled	True	False	True



Summary

- Asynchronous, threadless vs threaded tasks
- Task and Task<T>
- Error handling
- Continuations
- Composition
- State model

