SpinWait Struct

Reference

Definition

Namespace: System.Threading

Assembly: System.Threading.dll

Provides support for spin-based waiting.

C#

```
public struct SpinWait
```

Inheritance Object → ValueType → SpinWait

Examples

The following example shows how to use a SpinWait:

C#

```
using System;
using System.Threading;
using System.Threading.Tasks;

class SpinWaitDemo
{
    // Demonstrates:
    //      SpinWait construction
    //      SpinWait.SpinOnce()
    //      SpinWait.NextSpinWillYield
    //      SpinWait.Count
    static void Main()
    {
        bool someBoolean = false;
        int numYields = 0;

        // First task: SpinWait until someBoolean is set to true
        Task t1 = Task.Factory.StartNew(() =>
        {
            SpinWait sw = new SpinWait();
            while (!someBoolean)
            {
                // The NextSpinWillYield property returns true if
                // calling sw.SpinOnce() will result in yielding the
                // processor instead of simply spinning.
                if (sw.NextSpinWillYield) numYields++;
                sw.SpinOnce();
            }

            // As of .NET Framework 4: After some initial spinning,
SpinWait.SpinOnce() will yield every time.
            Console.WriteLine("SpinWait called {0} times, yielded {1} times",
sw.Count, numYields);
        });

        // Second task: Wait 100ms, then set someBoolean to true
        Task t2 = Task.Factory.StartNew(() =>
        {
            Thread.Sleep(100);
            someBoolean = true;
        });

        // Wait for tasks to complete
        Task.WaitAll(t1, t2);
    }
}
```

Remarks

SpinWait encapsulates common spinning logic. On single-processor machines, yields are always used instead of busy waits, and on computers with Intel processors employing Hyper-Threading technology, it helps to prevent hardware thread starvation. SpinWait encapsulates a good mixture of spinning and true yielding.

SpinWait is a value type, which means that low-level code can utilize SpinWait without fear of unnecessary allocation overheads. SpinWait is not generally useful for ordinary applications. In most cases, you should use the synchronization classes provided by the .NET Framework, such as Monitor. For most purposes where spin waiting is required, however, the SpinWait type should be preferred over the Thread.SpinWait method.

Properties

Count	Gets the number of times SpinOnce() has been called on this

	instance.
NextSpinWillYield	Gets whether the next call to SpinOnce() will yield the processor, triggering a forced context switch.

Methods

Reset()	Resets the spin counter.
SpinOnce()	Performs a single spin.
SpinOnce(Int32)	Performs a single spin and calls Sleep(Int32) after a minimum spin count.
SpinUntil(Func<Boolean>)	Spins until the specified condition is satisfied.
SpinUntil(Func<Boolean>, Int32)	Spins until the specified condition is satisfied or until the specified timeout is expired.
SpinUntil(Func<Boolean>, TimeSpan)	Spins until the specified condition is satisfied or until the specified timeout is expired.

Applies to

Product	Versions
.NET	Core 1.0, Core 1.1, Core 2.0, Core 2.1, Core 2.2, Core 3.0, Core 3.1, 5, 6, 7, 8
.NET Framework	4.0, 4.5, 4.5.1, 4.5.2, 4.6, 4.6.1, 4.6.2, 4.7, 4.7.1, 4.7.2, 4.8, 4.8.1
.NET Standard	1.0, 1.1, 1.2, 1.3, 1.4, 1.6, 2.0, 2.1
UWP	10.0
Xamarin.iOS	10.8
Xamarin.Mac	3.0

Thread Safety

While SpinWait is designed to be used in concurrent applications, it is not designed to be used from multiple threads concurrently. SpinWait members are not thread-safe. If multiple threads must spin, each should use its own instance of SpinWait.

See also

- SpinWait
- How to: Use SpinWait to Implement a Two-Phase Wait Operation