

LinksPlatform's Platform.Diagnostics Class Library

1.1 ./csharp/Platform.Diagnostics/Performance.cs

```
1 using System;
2 using System.Diagnostics;
3 using System.Runtime.CompilerServices;
4
5 namespace Platform.Diagnostics
6 {
7     /// <summary>
8     /// <para>Provides a set of methods for performance measurement.</para>
9     /// <para>Предоставляет набор методов для замера производительности.</para>
10    /// </summary>
11    public static class Performance
12    {
13        /// <summary>
14        /// <para>Measures the time it takes to execute the action by executing it and using
15        → <see cref="Stopwatch"/>.</para>
16        /// <para>Измеряет время, которое занимает выполнение действия, выполняя его и используя
17        → <see cref="Stopwatch"/>.</para>
18        /// </summary>
19        /// <param name="action"><para>The <see cref="Action"/> delegate.</para><para>Делегат
20        → <see cref="Action"/>.</para></param>
21        /// <returns>
22        /// <para>The time interval representing the time it took to execute the action.</para>
23        /// <para>Временной интервал, представляющий время, которое потребовалось для выполнения
24        → действия.</para>
25        /// </returns>
26        [MethodImpl(MethodImplOptions.AggressiveInlining)]
27        public static TimeSpan Measure(Action action)
28        {
29            var sw = Stopwatch.StartNew();
30            action();
31            sw.Stop();
32            return sw.Elapsed;
33        }
34    }
35 }
```

1.2 ./csharp/Platform.Diagnostics.Tests/PerformanceTests.cs

```
1 using System;
2 using Xunit;
3
4 namespace Platform.Diagnostics.Tests
5 {
6     /// <summary>
7     /// <para>
8     /// Represents the performance tests.
9     /// </para>
10    /// <para></para>
11    /// </summary>
12    public class PerformanceTests
13    {
14        /// <summary>
15        /// <para>
16        /// Tests that measure test.
17        /// </para>
18        /// <para></para>
19        /// </summary>
20        [Fact]
21        public void MeasureTest()
22        {
23            var result = 0;
24            var measurement = Performance.Measure(() => result = 1 + 1);
25            Assert.Equal(2, result);
26            Assert.True(measurement > TimeSpan.Zero);
27        }
28    }
29 }
```

Index

`./csharp/Platform.Diagnostics.Tests/PerformanceTests.cs`, 1

`./csharp/Platform.Diagnostics/Performance.cs`, 1