Caution

The SignalFx Instrumentation for .NET reached End of Support on February 21, 2025. The library has been archived and is no longer maintained.

New customers instrumenting the .NET ecosystem should use the Splunk Distribution of OpenTelemetry .NET. Existing customers should consider migrating to Splunk Distribution of OpenTelemetry .NET which offers similar capabilities. To learn how to migrate, see Migrate from the SignalFx .NET Instrumentation.

Manually instrument .NET applications for Splunk Observability Cloud

The SignalFx Instrumentation for .NET provides and registers an OpenTracing-compatible global tracer that you can use to instrument your applications manually for Splunk Observability Cloud. Custom or manual instrumentation can be helpful when you need to add custom attributes to spans, or need to generate spans manually.

Note

The SignalFx Instrumentation for .NET supports OpenTracing version 0.12.1 and higher.

To instrument your .NET application manually, follow these steps:

1. Add the OpenTracing dependency to your project:
```
<PackageReference Include="OpenTracing" Version="0.12.1" />
```

2. Obtain the `OpenTracing.Util.GlobalTracer` instance and create spans:
```
using OpenTracing;
using OpenTracing.Util;

namespace MyProject
{
  public class MyClass
  {
```

```
  public static async void MyMethod()
  {
    // Obtain the automatically registered OpenTracing.Util.GlobalTracer instance
    var tracer = GlobalTracer.Instance;

    // Create an active span that automatically becomes a child span of any existing span in this context
    using (IScope scope = tracer.BuildSpan("MyTracedFunctionality").StartActive(finishSpanOnDispose: true))
    {
      var span = scope.Span;
      span.SetTag("MyTag", "MyValue");
      span.Log("My Log Statement");

      var ret = await MyAppFunctionality();

      span.SetTag("FunctionalityReturned", ret.ToString());
    }
  }
}
       }
```