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 The library has been archived and is no longer maintained.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.

.NET instrumentation compatibility and requirements

Meet the following requirements to instrument .NET applications for Splunk Observability Cloud:

Ensure you are using supported .NET versions

The SignalFx Instrumentation for .NET supports the following .NET versions:

- Instrumentation for traces and metrics:
  - .NET 6.0
  - .NET Framework 4.6.2 and higher
- AlwaysOn Profiling:
  - .NET 6.0

Support for legacy .NET versions

Limited support is available for the following legacy versions of .NET:

- Instrumentation for traces and metrics:
  - .NET 7.x
  - .NET 5.x

    - .NET Core 3.1
    - .NET Framework 4.6.1
- AlwaysOn Profiling:
    - CPU Profiling: .NET Core 3.1, .NET 5.x, and .NET 7.x
    - Memory Profiling: .NET Core 5.x and .NET 7.x

Supported libraries

The SignalFx Instrumentation for .NET instruments the following libraries:

Library	Instrumentation ID
`Aerospike.Client`	`Aerospike`
ASP.NET 4.x	`AspNet`
ASP.NET Core	`AspNetCore`
ASP.NET MVC	`AspNetMvc`
ASP.NET Web API 2	`AspNetWebApi2v`
`AWSSDK.Core` (Experimental)	`AwsSdk`
`AWSSDK.SQS` (Experimental)	`AwsSqs`
`Confluent.Kafka`	`Kafka`
`CouchbaseNetClient` (Experimental)	`Couchbase`
`Elasticsearch.Net`	`ElasticsearchNetv`

GraphQL	`GraphQL`
gRPC	`Grpc`
`Microsoft.Data.SqlClient` and `System.Data.SqlClient`	`SqlClient`
`Microsoft.Extensions.Logging.Abstractions`	`ILOgger`
`Microsoft.Azure.Cosmos` (Experimental)	`CosmosDb`
`Microsoft.Azure.WebJobs` (Experimental)	`AzureFunctions`
`Microsoft.ServiceFabric.Services.Remoting` (Experimental)	`ServiceRemoting`
`Microsoft.VisualStudio.TestPlatform` (Experimental)	`MsTestV2`
`MongoDB.Driver.Core`	`MongoDb`
`MySql.Data`	`MySql`
Npgsql	`Npgsql`
`NUnit` (Experimental)	`NUnit`
`Oracle.ManagedDataAccess`	`Oracle`
`RabbitMQ.Client`	RabbitMQ
`ServiceStack.Redis`	`ServiceStacksRedis`
SQLite	`Sqlite`

`StackExchange.Redis`	`StackExchangeRedis`
`System.Net.Http.CurlHandler`	`CurlHandler`
`System.Net.Http.MessageHandler`	`HttpMessageHandler`
`System.Net.Http.SocketsHandler`	`HttpSocketsHandler`
`System.Net.Http.WinHttpHandler`	`WinHttpHandler`
`System.Net.WebRequest`	`WebRequest`
`System.Messaging` (Experimental)	`Msmq`
Windows Communication Foundation (WCF)	`Wcf`
`xunit` (Experimental)	`XUnit`

To instrument the `System.Net.Http.HttpClient` library, you must instrument the following group of libraries:

- `System.Net.Http.CurlHandler`
- `System.Net.Http.MessageHandler`
- `System.Net.Http.SocketsHandler`
- `System.Net.Http.WinHttpHandler`

Install and configure the Splunk Distribution of OpenTelemetry Collector

The SignalFx Instrumentation for .NET exports application traces and spans to the Splunk Distribution of OpenTelemetry Collector, which also collects system metric data and logs, including profiling data.

To send application traces and spans to Splunk Observability Cloud, install the Splunk Distribution of OpenTelemetry Collector for your platform. The following distributions are available:

- Splunk OTel Collector for Linux. See Install the Collector for Linux with the installer script.
- Splunk OTel Collector for Windows. See Install the Collector for Windows with the installer script.
- Splunk OTel Collector for Kubernetes. See Install the Collector for Kubernetes using Helm.

Note

The OTel Collector is not required when instrumenting Azure App Service applications. See Instrument your application in Azure App Service.