DependencyInjectionContainerBenchmarker

Contents

Summary

DependencyInjectionContainerBenchmarker is a demonstration application showing the concept of benchmarking third-party dependency injection (DI) containers.

To benchmark a DI container, the application firstly creates 2,000 C# classes dynamically, each implementing a dynamically created C# interface.

It then
(a.) registers the 2,000 dynamically created types with the DI container, each against its interface;
(b.) retrieves the 2,000 dynamically created types from the DI container, each by its registered interface.

The time taken to carry out these two operations is then output as the benchmark for the DI container.

Executing the application

To run this application, use the following command-line:

DependencyInjectionContainerBenchmarker[.exe] *container-name*

where container-name is the name of the DI container to be benchmarked. (For supported DI containers, see the section ‘Supported DI containers,’ *infra.*)

Adding a DI container to be benchmarked

To benchmark a particular third-party DI container, a project must be added to the ContainerAbstractions solution folder, containing a class which inherits the ContainerAbstraction abstract class. The inheritor of ContainerAbstraction acts as a

standardized 'layer' over the DI container to be benchmarked. This can then be benchmarked via the IContainerAbstraction interface (which the ContainerAbstraction abstract class implements).

In addition,

(a.) a project reference to the new container abstraction project should be added to the DependencyInjectionContainerBenchmarker executable;

(b.) a NuGet package reference to the DI container should be added to the DependencyInjectionContainerBenchmarker executable (so that the DI container's DLL is available for the application to create an instance of the DI container via the IContainerAbstraction interface);

(c.) The following class/enumeration types in the DependencyInjectionContainerBenchmarker project should be amended to add the new DI container:

- Enumerations\ContainerType
- Helpers\CommandLineParser
- Helpers\ContainerAbstractionFactory

Supported DI containers

The application currently supports the following DI containers:—

DI Container	***container-name* on command line**
Simple Injector	SimpleInjector
Unity	Unity

Notes on implementation

- Antivirus software may treat this application as suspicious when executed, owing to its generation of dynamic types. This is expected and nothing to be concerned about.

- The generation of dynamic types using System.CodeDom is apparently not supported on .NET 8.0 or 9.0, on which I initially attempted it. It works on .NET Framework 4.8, which is why the DependencyInjectionContainerBenchmarker executable and the DependencyInjectionContainerBenchmarker.Application class library target this framework. The other projects (including the container abstraction implementations) target .NET Standard 2.0.

Graham Harter
6th March 2025