

The Ultimate List of .NET Dependency Injection Frameworks

by Claudio Bernasconi | Jan 24, 2019 | C# | 10 comments



Dependency Injection. Some love it, some hate it. For some developers it might be an anti-pattern, unnecessary, other developers cannot think about building applications without it.

I believe that there are use cases where Dependency Injection frameworks are a good choice. Therefore, I created a few videos about Dependency Injection on my [YouTube channel](#). Anyway, Dependency Injection frameworks are still around in 2019.

There is no overview about the current state of Dependency Injection frameworks available on the internet. Therefore, I decided to create a list as a reference.

The goal of this overview is to provide facts and help you narrow down on the options for your project. It's not meant to be a guide on which framework to choose or to give opinionated recommendations.

The following list contains all available Dependency Injection frameworks for the .NET platform in alphabetical order beginning with the actively maintained and developed frameworks ending with the *inactive* and *deprecated* frameworks.

[Diagram: Comparison of avg. daily downloads per framework](#)

Table of contents

- [Autofac](#)
- [Castle Windsor](#)
- [Lamar](#)
- [LightInject](#)
- [Ninject](#)
- [SimpleInjector](#)
- [Spring.NET](#)
- [Unity](#)

- [LinFu](#) (*inactive*)
- [Managed Extensibility Framework \(MEF\)](#) (*abandoned / deprecated*)
- [PicoContainer.NET](#) (*abandoned / deprecated*)
- [S2Container.NET](#) (*abandoned / deprecated*)
- [StructureMap](#) (*abandoned / deprecated*)

Autofac



Autofac is an additive Inversion of Control container for .NET Core, ASP.NET Core, .NET 4.5.1+, Universal Windows apps, and more. Its core features are the modularity system and the fluent API for dependency registrations.

As it was written when the .NET Framework 3.5 was released, it was the first Dependency Injection framework to support lambda registrations. According to the original creator [Nicholas Blumhardt](#) the inspiration for this feature came from Needle which is a IoC container for [Ruby](#).

Website	https://www.autofac.org/
Platforms	.NET Framework 4.5, .NET Core, Silverlight 5, Universal Windows Platform Apps (UWP), Windows Phone 8
GitHub	2300+ Stars, 550+ Forks
First released	2007
NuGet	19M+ total downloads, 6.5K+ downloads per day (avg)

Additional Features	Interceptors, Concurrency Handling, Multitenant support, various (test) framework integrations
Documentation	Documentation
Development	active

Hint: If you want to get into Dependency Injection using Autofac, check out my [YouTube channel](#) where I show the basics of Dependency Injection using Autofac.

Castle Windsor



According to their project website, "Castle Windsor is a best of breed, mature Inversion of Control container available for .NET."

Website	http://www.castleproject.org/projects/windsor/
Platforms	.NET Framework 4.5, .NET Standard 1.6 (with a bunch of missing features, see project page for more details)
GitHub	900+ Stars, 350+ Forks
First released	2004

NuGet	4.4M+ total downloads, 1.5K downloads per day (avg)
Additional Features	Aspect-oriented programming (AOP), Proxies, and Interceptors (using Castle.DynamicProxy)
Documentation	Documentation
Development	active

According to [Jonathon Rossi](#), one of the active maintainers, they plan to add .NET Standard 2.1 support soon since Reflection.Emit has been restored to .NET Standard which is needed for DynamicProxy related features. Some of the unnecessary legacy features will not be supported.

Lamar

Lamar is the new kid on the block. Its initial commit date is March 2018 according to the [GitHub](#) repository. The idea behind Lamar is to create a successor to StructureMap. According to the project website, its focus is the performance.

Because Lamar supports most of StructureMap's IContainer syntax, Lamar is an interesting opportunity if a project is currently using StructureMap and should be moved to a more modern, faster framework. As it's one of the newest frameworks it's also worth considering to be used in new projects.

Lamar provides [integration to the ASP.NET core Dependency Injection](#) mechanism.

Website	https://jasperfx.github.io/lamar/
Platforms	.NET Framework 4.6+, .NET Standard 2.0

GitHub	140+ Stars, 20+ Forks
First released	2018
NuGet	25K+ total downloads, 75+ downloads per day (avg)
Additional Features	Interceptors using Decorators, Diagnostics
Documentation	Documentation
Development	active

LightInject

LightInject proclaims to be an ultra-lightweight IoC container for the .NET platform. This does not mean that the feature set is limited. LightInject provides different lifetime scopes, explicit and implicit service registration, as well as lazy dependency resolution.



It comes with many framework integrations and makes use of modern .NET platform features to provide a good developer experience. LightInject also provides clean and understandable documentation featuring code examples and therefore makes it easy to get started.

Website	https://www.lightinject.net/
Platforms	.NET Framework 4.5.2+, .NET Standard 1.1
GitHub	340+ Stars, 100+ Forks

First released	2011
NuGet	650K+ total downloads, 240+ downloads per day (avg)
Additional Features	Integrations with SignalR, Nancy, ASP.NET core, xUnit
Documentation	Documentation
Development	active

Ninject



Ninject is one of the well-known .NET Dependency Injection frameworks. Ninject claims to make your software much easier to change after its initial release because Ninject is focused on simplicity and ease of use.

Ninject has a big community which developed many extensions to the core Ninject project including MVC support, WCF support and many more.

Website	http://www.ninject.org
Platforms	.NET Framework 3.0+, Silverlight, Windows Phone 7, Mono, .NET Standard 2.0
GitHub	2100+ Stars, 500+ Forks
First released	2008
NuGet	8.4M+ total downloads, 2.8K+ downloads per day (avg)
Additional Features	MVC support, WCF support, interceptors, conventions, named scopes, Logging,

	message brokers, Quartz, and many more.
Documentation	Documentation
Development	active

SimpleInjector

Simple Injector is an easy-to-use Dependency Injection (DI) library. Simple Injector not only wants to be simple to use but is also optimized for performance and concurrent use. Its the goal of the project to be an alternative for developers who want to focus on the important stuff while SimpleInjector is doing its job.



Website	https://simpleinjector.org
Platforms	.NET Framework 4.0+, .NET Standard Mono, Silverlight, Windows Phone, UWP
GitHub	700+ Stars, 100+ Forks
First released	2010
NuGet	4.3M+ total downloads, 1.6K+ downloads per day (avg)
Additional Features	(web) framework integrations, Wcf integration
Documentation	Documentation
Development	active

Spring.NET

According to their project website, “Spring.NET provides comprehensive infrastructural support for developing enterprise .NET applications. (...) The design of Spring.NET is based on the Java version of the Spring Framework.”

Spring.NET contains various NuGet packages with additional features beyond dependency injection such as aspect-oriented programming, ADO.NET and NHibernate adapters, test integrations and much more.



Website	http://springframework.net/
Platforms	.NET Framework 4.5.2, .NET Standard 2.0
GitHub	500+ Stars, 250+ Forks
First released	2005
NuGet	310K+ total downloads, 110+ downloads per day (avg)
Additional Features	aspect-oriented programming AOP, data adapters (ADO.Net, NHibernate), Web extensions, NUnit, MsTest, Messaging, Quartz, Velocity.
Documentation	Documentation
Development	active

Unity

The Unity Container (Unity) is a lightweight, extensible dependency injection container. It facilitates building loosely coupled applications and provides various advantages for developers. It supports registration by convention and type interception.



Unity was created and developed by Microsoft and [handed off to the community](#) in 2015.

Website	https://unitycontainer.github.io/
Platforms	.NET Framework 4.0+, .NET Core, .NET Standard 1.0 and 2.0
GitHub	1000+ Stars, 250+ Forks
First released	2008
NuGet	14.9M+ total downloads, 5.1K+ downloads per day (avg)
Additional Features	type interception, framework integrations
Documentation	Documentation
Development	active

LinFu

LinFu ("Language INdependent Features Underneath [.NET].") is a small project founded by [Philip Laureano](#) in 2007. It is a framework that adds mixins, inversion of control and other language features to the Common

Language Runtime (CLR).

Website	https://github.com/philiplaureano/LinFu
Platforms	.NET Framework 2.0+
GitHub	180+ Stars, 30+ Forks
First released	2007
NuGet	310K+ total downloads, 110+ downloads per day (avg)
Additional Features	Mixins
Documentation	Documentation
Development	inactive

Managed Extensibility Framework (MEF)

The following description is taken from the official .NET framework [documentation article about MEF](#) provided by Microsoft:

“The Managed Extensibility Framework or MEF is a library for creating lightweight, extensible applications. It allows application developers to discover and use extensions with no configuration required. It also lets extension developers easily encapsulate code and avoid fragile hard dependencies. MEF not only allows extensions to be reused within applications but across applications as well.”

Website	https://github.com/MicrosoftArchive/mef
Platforms	.NET Framework 4.0+

GitHub	60+ Stars, 20+ Forks
First released	2010
NuGet	not available
Documentation	Documentation
Development	deprecated

PicoContainer.NET

Pico Container.NET is a lightweight dependency injection framework providing only the necessary features, no overhead. The project is no longer continued.

Website	http://picocontainer.com/
Platforms	.NET Framework
GitHub	90+ Stars, 20+ Forks
First released	2004
NuGet	not available
Documentation	Documentation
Development	abandoned, last release on October, 8th 2014.

S2Container.NET

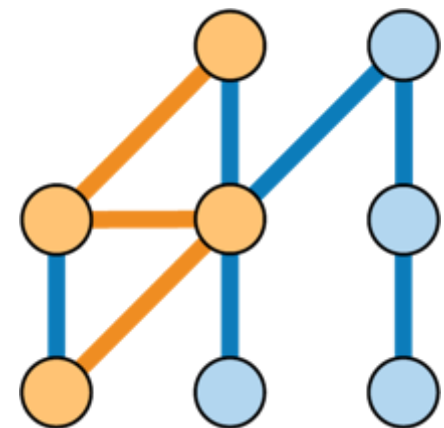
S2Container.NET is another Dependency Injection framework that was ported from a Java library back in 2005. It is not widely-known and development was stopped in 2015.

Website	https://github.com/MicrosoftArchive/mef
Platforms	.NET Framework 2.0
GitHub	not available
First released	2005
NuGet	not available
Documentation	Documentation
Development	abandoned , last release 1.4.1 in 2015.

StructureMap

From the project website: "StructureMap is the oldest, continuously used IoC/DI container for .NET dating back to its first public release in June 2004 on .NET 1.1."

As the grand-father of dependency injection frameworks, StructureMap offers feature-wise everything developers expect from a DI container: A modular configuration, lazy dependency resolution, resolution of IEnumerable types, diagnostics, and child container support. These features listed here are only a few selected features.



Meanwhile, [StructureMap has been sunsetted](#) as announced by [Jeremy Miller](#) on January 29th, 2018 and the maintainers recommend using [Lamar](#) as a possible replacement.

Website	http://structuremap.github.io/
Platforms	.NET Framework 4.5, .NET Standard 1.6, Windows Phone (up to 8.1)
GitHub	800+ Stars, 250+ Forks
First released	2004
NuGet	5.3M+ total downloads, 1.8K downloads per day (avg)
Documentation	Documentation
Development	deprecated

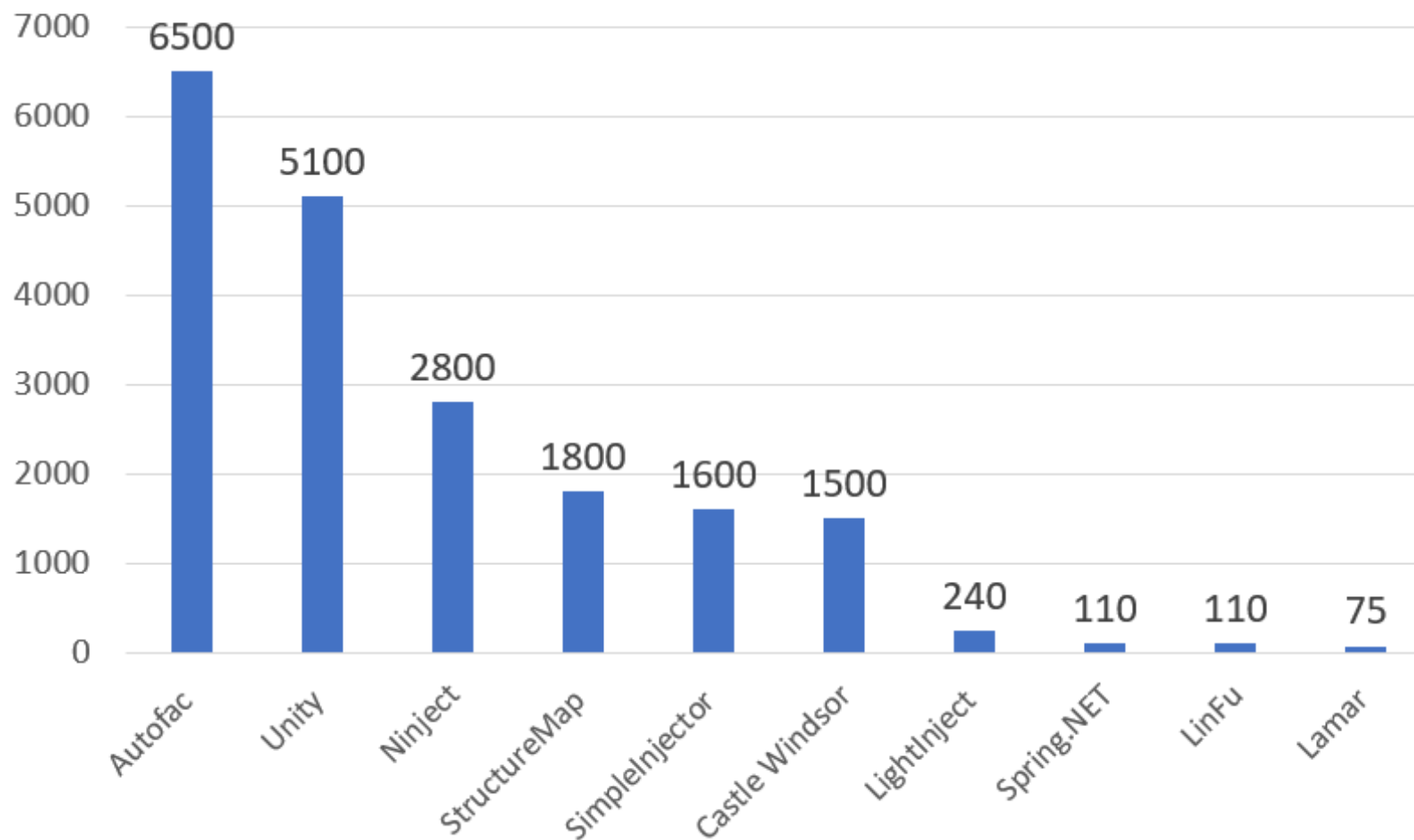
Framework comparison

The following diagram shows the number of daily downloads the frameworks listed on NuGet.com get on average.

Remember, this is just a metric to show the current state, and it does not mean that the most popular framework is always the best choice for your project. Also, new projects that have not been there for a long time (e.g., Lamar) do not have that many downloads yet.

The numbers were taken from Nuget.com on January 24th, 2019.

Downloads per day (avg)



If there is a framework missing on this list or do you believe that some information is wrong? Please write it in the comments or [DM me on Twitter](#) so that this list always stays current and complete.

Other Resources

- [IoC Performance comparison](#) by Daniel Palme



Claudio Bernasconi

Claudio Bernasconi is a Senior Software Engineer, Microsoft certified technology specialist, blogger, technology enthusiast and founder of CarParking Schweiz.

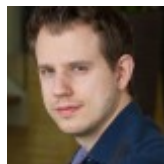
10 Comments



Scott on 29.01.2019 at 22:16

MEF is not a dependency injection framework. It's for discovering components in dynamically loaded assemblies and putting weird attributes all over stuff.

[Reply](#)



Claudio Bernasconi on 30.01.2019 at 10:38

Hi Scott! I really appreciate your feedback and your opinion on the topic. MEF was indeed created for component discovery as you stated in your comment. I have been working for a company which used MEF for more of a DI approach. It's the reason I included it to the list.

[Reply](#)

Sometimes tools are used when there are better alternatives. I hope you don't mind that MEF will remain on the list for the sake of completeness?



Scott on 01.02.2019 at 22:57

That's funny because that's exactly how I encountered it. I found myself in a place where it is not only used for DI (sort of, not really) but if you use anything else they reject it and say you can only use MEF.

Sorry if I made it sound like I was telling you what to put on your list.

Here's something I wrote after they forced me to use MEF: <http://scotthannen.org/blog/2018/12/16/liberation-of-pure-di-mef-is-not-ioc.html>

Reply



Roman on 06.02.2019 at 09:44

What about Dryloc? It shows great performance and has compile time versiobln

Reply

Jason on 13.02.2019 at 00:21

Reply



Why not mention the ASP.NET Core DependencyInjection Framework?

<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-2.2>



Jan D Narkiewicz on 19.04.2019 at 18:03

Great article. Comprehensive. Well organized. You made my job easier.

Reply



Claudio Bernasconi on 14.05.2019 at 01:08

Thank you very much for your kind feedback.

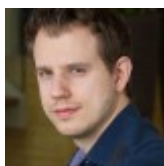
Reply



Jaishree Kulkarni on 21.06.2019 at 01:44

Very good Article. Information very well organised and clearly explains the advantages and disadvantages in brief.

Reply



Claudio Bernasconi on 23.06.2019 at 00:46

Thank you very much for your kind feedback.

Reply



Miles Davies on 07.08.2019 at 10:35

MEF is part of the NET framework and is definitely still active. You can download it on nuget using the `system.componentmodel.composition` assembly name

Reply

Submit a Comment

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Submit Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

Claudio Bernasconi



Categories

Bloggng (4)

C# (32)

Clean Code (7)

Cloud Development (8)
Continuous Integration (3)
Java (3)
JavaScript (2)
Learning (13)
Mobile Development (1)
Personal (8)
Reviews (7)
Source Control (5)
Visual Studio (9)
YouTube (11)

Most popular post

When To Use IEnumerable, ICollection, IList And List

Most commented post

Microsoft 70-483 Exam Preparation Guide

Recent Posts

Choose Your Favorite .NET Show!

How to Write Your First Azure Function App

7 Reasons Why Serverless Is the Future

Archives

2019 (30)

2018 (5)

2017 (3)

2016 (1)

2015 (4)

2014 (6)

2013 (20)

2012 (3)

2010 (1)

2008 (1)

Meta

Log in

Entries [RSS](#)

Comments [RSS](#)

[WordPress.org](#)



© 2008-2018 **Claudio Bernasconi** | Designed by **Elegant Themes** | Powered by **WordPress**