

# Which .NET Dependency Injection frameworks are worth looking into?

[closed]

Asked 16 years, 4 months ago   Modified 3 years ago   Viewed 185k times



409



As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

Closed 12 years ago.

Which C#/.NET Dependency Injection frameworks are worth looking into? And what can you say about their complexity and speed.

c#

.net

dependency-injection

inversion-of-control

Share

Improve this question

Follow

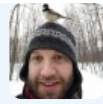
edited Jul 28, 2010 at 15:02



John Saunders

162k ● 26 ● 249 ● 402

asked Aug 21, 2008 at 21:56



pbreault

14.5k ● 19 ● 46 ● 38

- 
- 14 [IoC Container Benchmark - Performance comparison](#) has performance and features comparison tables for 20+ products and keep them up-to-date. It recommends [Simple Injector](#) – [Michael Freidgeim](#) Mar 22, 2013 at 22:07 ✎
- 

I appreciate Ninject & Maestro. I'm happy that the top rated answer reffer Ninject as "an absolute pleasure"

– [Razvan Dumitru](#) Jun 26, 2014 at 0:28

---

- 7 I'd stress that before you look into IoC containers, you should manage without one. Do your wiring and injection manually, it's not really complicated or requiring of a lot of code at all unless you have some really complex or large systems. When you feel that you understand it and that it is becoming combersome you will be better fitted to understand what kind of framework wuold help you. but there is no rule that says that a project MUST have an IoC container just to be able to do DI properly. – [sara](#) May 3, 2016 at 6:28
- 

- 2 Start with none. Create required objects graph manually. Different frameworks has different approaches which depends on your needs. Before choosing one you need to recognize which features you need most. – [Fabio](#) Jan 25, 2019 at 20:53 ✎
- 

- 1 Since dotnetcore is the future, you might as well stick with its built in DI support. It's very good and highly customizable. – [ATL\\_DEV](#) Sep 20, 2019 at 15:42
- 

12 Answers

Sorted by:

Highest score (default)



**edit** (not by the author): There is a comprehensive list of IoC frameworks available at



- [Castle Windsor](#) - Castle Windsor is best of breed, mature Inversion of Control container available for .NET and Silverlight
- [Unity](#) - Lightweight extensible dependency injection container with support for constructor, property, and method call injection
- [Autofac](#) - An addictive .NET IoC container
- [Dryloc](#) - Simple, fast all fully featured IoC container.
- [Ninject](#) - The ninja of .NET dependency injectors
- [Spring.Net](#) - Spring.NET is an open source application framework that makes building enterprise .NET applications easier
- [Lamar](#) - A fast IoC container heavily optimized for usage within ASP.NET Core and other .NET server side applications.
- [LightInject](#) - A ultra lightweight IoC container
- [Simple Injector](#) - Simple Injector is an easy-to-use Dependency Injection (DI) library for .NET 4+ that supports Silverlight 4+, Windows Phone 8, Windows 8 including Universal apps and Mono.
- [Microsoft.Extensions.DependencyInjection](#) - The default IoC container for ASP.NET Core applications.
- [Scrutor](#) - Assembly scanning extensions for Microsoft.Extensions.DependencyInjection.

- [VS MEF](#) - Managed Extensibility Framework (MEF) implementation used by Visual Studio.
- [TinyIoC](#) - An easy to use, hassle free, Inversion of Control Container for small projects, libraries and beginners alike.
- [Stashbox](#) - A lightweight, fast and portable dependency injection framework for .NET based solutions.

Original answer follows.

---

I suppose I might be being a bit picky here but it's important to note that DI (Dependency Injection) is a programming pattern and is facilitated by, but does not require, an IoC (Inversion of Control) framework. IoC frameworks just make DI much easier and they provide a host of other benefits over and above DI.

That being said, I'm sure that's what you were asking. About IoC Frameworks; I used to use [Spring.Net](#) and [CastleWindsor](#) a lot, but the real pain in the behind was all that pesky XML config you had to write! They're pretty much all moving this way now, so I have been using [StructureMap](#) for the last year or so, and since it has moved to a fluent config using strongly typed generics and a registry, my pain barrier in using IoC has dropped to below zero! I get an absolute kick out of knowing now that my IoC config is checked at compile-time (for the most part) and I have had nothing but joy with StructureMap and its speed. I won't say that the others

were slow at runtime, but they were more difficult for me to setup and frustration often won the day.

## Update

I've been using [Ninject](#) on my latest project and it has been an absolute pleasure to use. Words fail me a bit here, but (as we say in the UK) this framework is 'the Dogs'. I would highly recommend it for any green fields projects where you want to be up and running quickly. I got all I needed from a [fantastic set of Ninject screencasts](#) by Justin Etheredge. I can't see that retrofitting Ninject into existing code being a problem at all, but then the same could be said of [StructureMap](#) in my experience. It'll be a tough choice going forward between those two, but I'd rather have competition than stagnation and there's a decent amount of healthy competition out there.

Other IoC screencasts can also be found [here on Dimecasts](#).

Share Improve this answer

Follow

edited Dec 6, 2021 at 7:49



Péter Csajtai

898 ● 6 ● 9

answered Aug 21, 2008 at 22:29



RobertTheGrey

8,685 ● 5 ● 33 ● 45

---

1 It's great to see a quality, well thought out answer rise about the one liners. You convinced me to check out StructureMap.  
– [Chris Marasti-Georg](#) Oct 30, 2008 at 15:13

---

7 Nice but to be fair - Windsor has a very nice fully-fledged fluent interface now as well. – [Krzysztof Kozmic](#) Dec 1, 2009 at 15:01

---

Could you explain what are the "host of other benefits" IoC frameworks provide besides easy implementation of DI?

– [fearofawhackplanet](#) Feb 14, 2012 at 9:45 ✎

---

@fearofawhackplanet one of the only things an IoC container can do that can't be done with *poor man's DI* is **Interception**.

Good summary here: [kenneth-truyers.net/2013/05/16/...](http://kenneth-truyers.net/2013/05/16/...)

– [Mathieu Guindon](#) Jul 25, 2013 at 16:54

---

2 I have read a couple of sources that say that Ninject is the slowest [palmmedia.de/blog/2011/8/30/...](http://palmmedia.de/blog/2011/8/30/...) and [stackoverflow.com/questions/4581791/...](http://stackoverflow.com/questions/4581791/...) – [Luke T O'Brien](#) Dec 6, 2016 at 9:34

---



It depends on what you are looking for, as they each have their pros and cons.

77



1. **Spring.NET** is the most mature as it comes out of Spring from the Java world. Spring has a very rich set of framework libraries that extend it to support Web, Windows, etc.

2. **Castle Windsor** is one of the most widely used in the .NET platform and has the largest ecosystem, is highly configurable / extensible, has custom lifetime management, AOP support, has inherent NHibernate

support and is an all around awesome container.

Windsor is part of an entire stack which includes Monorail, Active Record, etc. NHibernate itself builds on top of Windsor.

3. **Structure Map** has very rich and fine grained configuration through an internal DSL.
4. **Autofac** is an IoC container of the new age with all of it's inherent functional programming support. It also takes a different approach on managing lifetime than the others. Autofac is still very new, but it pushes the bar on what is possible with IoC.
5. **Ninject** I have heard is more bare bones with a less is more approach (heard not experienced).
6. The biggest discriminator of **Unity** is: it's from and supported by Microsoft (p&p). Unity has very good performance, and great documentation. It is also highly configurable. It doesn't have all the bells and whistles of say Castle / Structure Map.

So in summary, it really depends on what is important to you. I would agree with others on going and evaluating and seeing which one fits. The nice thing is you have a nice selection of donuts rather than just having to have a jelly one.

Share Improve this answer

Follow

edited Jun 27, 2015 at 6:05



shA.t

16.9k ● 5 ● 57 ● 119

answered Oct 22, 2008 at 18:29



Glenn Block

8,445 ● 1 ● 34 ● 34

- 
- 5 Autofac is actually not so new, it is older than Unity :)  
– [Nicholas Blumhardt](#) Jan 7, 2010 at 3:13
- 
- 1 Did I say it was newer than Unity? I said it is of the new age...ie I meant it's functional nature. OK, what I said it's still very new, what I meant though was it's nature not that IT was new. :-)  
– [Glenn Block](#) Jan 17, 2010 at 18:54
- 
- 2 @Krzysztof - I find Unity dead easy (at least when configured fluently in code). What did you find painful? – [TrueWill](#) Jan 19, 2011 at 23:24
- 
- 1 @Krzysztof - [Using the UnityContainer Fluent Interface](#)  
– [TrueWill](#) Jun 18, 2011 at 17:45 ✎
- 
- 6 Here's an interesting performance benchmark:  
[palmmedia.de/Blog/2011/8/30/...](http://palmmedia.de/Blog/2011/8/30/...) – [Steven](#) Dec 13, 2011 at 10:51
- 



36

Autofac. <https://github.com/autofac/Autofac> It is really fast and pretty good. Here is a link with comparisons (made after Ninject fixed a memory leak issue).



<http://www.codinginstinct.com/2008/05/ioc-container-benchmark-rerevisted.html>



Share Improve this answer

Follow

edited Feb 23, 2018 at 9:15



Nick N.

13.5k ● 9 ● 61 ● 77

answered Aug 22, 2008 at 0:35





Pete

4,041 ● 3 ● 32 ● 31

- 
- 1 I do like autofac but it's certainly not fast. As Steven linked above: [palmmedia.de/Blog/2011/8/30/...](http://palmmedia.de/Blog/2011/8/30/...) – Serj Sagan Aug 10, 2016 at 21:11
- 



18



Ninject is great. It seems really fast, but I haven't done any comparisons. I know Nate, the author, did some comparisons between Ninject and other DI frameworks and is looking for more ways to improve the speed of Ninject.



I've heard lots of people I respect say good things about StructureMap and CastleWindsor. Those, in my mind, are the big three to look at right now.

Share Improve this answer

answered Aug 21, 2008 at 22:00

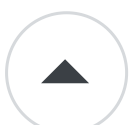
Follow



ScottKoon

3,493 ● 6 ● 28 ● 29

- 
- 1 Ninject is popular but it is certainly NOT fast. It is actually one of the slowest out there: [palmmedia.de/Blog/2011/8/30/...](http://palmmedia.de/Blog/2011/8/30/...) – Serj Sagan Aug 10, 2016 at 21:11
- 



14

I use [Simple Injector](#):

Simple Injector is an easy, flexible and fast dependency injection library that uses best



practice to guide your solutions toward the pit of success.



Share Improve this answer

Follow

edited Oct 20, 2014 at 11:18



Steven

172k ● 25 ● 346 ● 447

answered Jan 6, 2010 at 11:37




Steven

322 ● 3 ● 2

- 
- 4 Beware some consider the service locator an anti-pattern, including someone who used it for some time and even wrote a library for it:

[blog.ploeh.dk/2010/02/03/ServiceLocatorisanAnti-Pattern](http://blog.ploeh.dk/2010/02/03/ServiceLocatorisanAnti-Pattern)

– Philippe Sep 10, 2013 at 19:37

- 
- 4 Steven is also the creator of Simple Injector... so this is kind of a spammy answer... especially that he used an alt accounts to do so... Simple Injector is pretty fast though: [palmmedia.de/Blog/2011/8/30/...](http://palmmedia.de/Blog/2011/8/30/...) – Serj Sagan Aug 10, 2016 at 21:13 
-



8



I'm a huge fan of Castle. I love the facilities it also provides beyond the IoC Container story. It really simplifies using NHibernate, logging, AOP, etc. I also use [Binsor](#) for configuration with Boo and have really fallen in love with Boo as a language because of it.

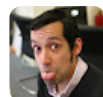


Share Improve this answer

answered Sep 16, 2008 at 9:39



Follow



[Chris Canal](#)

4,865 ● 9 ● 37 ● 45



5



I spent the better part of a day struggling without success to get the simplest Spring.NET example working. Could never figure out how to get it to find my assembly from the XML file. In about 2 hours, on the other hand, I was able to get Ninject working, including testing integration with both NUnit and MSTest.



Share Improve this answer

answered Mar 22, 2011 at 17:35



Follow



[outinsun](#)

51 ● 1 ● 1



4



I've used [Spring.NET](#) in the past and had great success with it. I never noticed any substantial overhead with it, though the project we used it on was fairly heavy on its own. It only took a little time reading through the [documentation](#) to get it set up.



Share Improve this answer

answered Aug 21, 2008 at 22:06

Follow



Jason Sparks  
1,306 ● 11 ● 11



4

I can recommend Ninject. It's incredibly fast and easy to use but only if you don't need XML configuration, else you should use Windsor.



Share Improve this answer

answered Aug 22, 2008 at 0:40

Follow



Julio César  
13.3k ● 10 ● 40 ● 46



1 It is NOT fast! It is one of the slowest DI frameworks out there! [palmmedia.de/Blog/2011/8/30/...](http://palmmedia.de/Blog/2011/8/30/...) – Serj Sagan Aug 10, 2016 at 21:15

@Serj Sagan To be fair, those results are ancient (6 years old now) – Timothy Kanski Mar 10, 2017 at 19:46

3 [akshaysura.com/2016/08/31/...](http://akshaysura.com/2016/08/31/...) – Timothy Kanski Mar 10, 2017 at 19:52



2

The great thing about C# is that it is following a path beaten by years of Java developers before it. So, my advice, generally speaking when looking for tools of this nature, is to look for the solid Java answer and see if there exists a .NET adaptation yet.



So when it comes to DI (and there are so many options out there, this really is a matter of taste) is [Spring.NET](http://Spring.NET). Additionally, it's always wise to research the people

behind projects. I have no issue suggesting SourceGear products for source control (outside of using them) because I have respect for Eric Sink. I have seen Mark Pollack speak and what can I say, the guy just gets it.

In the end, there are a lot of DI frameworks and your best bet is to do some sample projects with a few of them and make an educated choice.

Good luck!

Share Improve this answer

answered Aug 21, 2008 at 22:07

Follow



[Ian Patrick Hughes](#)

10.4k ● 3 ● 32 ● 37



2

I think a good place to start is with Ninject, it is new and has taken into account alot of fine tuning and is really fast. Nate, the developer, really has a great site and great support.



Share Improve this answer

answered Aug 22, 2008 at 0:37

Follow



[Rob Bazinet](#)

1,020 ● 1 ● 8 ● 12



1 Why are you saying it is fast??? It is one of the slowest that there is! [palmmedia.de/Blog/2011/8/30/...](http://palmmedia.de/Blog/2011/8/30/...) – [Serj Sagan](#) Aug 10, 2016 at 21:16



Spring.Net is quite solid, but the documentation took some time to wade through. Autofac is good, and while

2

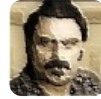
.Net 2.0 is supported, you need VS 2008 to compile it, or else use the command line to build your app.



Share Improve this answer

Follow

answered Sep 28, 2008 at 14:01



[David Robbins](#)

10k ● 7 ● 54 ● 83

---