What are the correct version numbers for C#?

Asked 10 years, 11 months ago Active 1 month ago Viewed 329k times

2389

593

What are the correct version numbers for C#? What came out when? Why can't I find any answers about ***C# 3.5***?

This question is primarily to aid those who are searching for an answer using an incorrect version number, e.g. ***C# 3.5***. The hope is that anyone failing to find an answer with the wrong version number will find *this* question and then search again with the right version number.

c# .net visual-studio .net-framework-version compiler-version

edited Dec 17 '17 at 17:09

community wiki
33 revs, 27 users 24%
Jon Skeet

71 This is one of a good source to understand everything. en.wikipedia.org/wiki/C_Sharp_(programming_language) – user725388 Dec 2 '12 at 8:21

1 Shouldn't that second paragraph be in a comment instead of the question, since it's not part of the question – TankorSmash Jan 11 '17 at 17:48

18 @TankorSmash: I think it's sufficiently important as the context of the question that it's worth keeping where it is. IMO, of course. – Jon Skeet Jan 11 '17 at 17:53

9 Answers

2748

C# language version history:

These are the versions of C# known about at the time of this writing:

- **C# 1.0** released with .NET 1.0 and VS2002 (January 2002)

- **C# 2.0** released with .NET 2.0 and VS2005 (November 2005). Major new features: generics, anonymous methods, nullable types, iterator blocks
- **C# 3.0** released with .NET 3.5 and VS2008 (November 2007). Major new features: lambda expressions, extension methods, expression trees, anonymous types, implicit typing (`var`), query expressions
- **C# 4.0** released with .NET 4 and VS2010 (April 2010). Major new features: late binding (`dynamic`), delegate and interface generic variance, more COM support, named arguments, tuple data type and optional parameters
- **C# 5.0** released with .NET 4.5 and VS2012 (August 2012). Major features: async programming, caller info attributes. Breaking change: loop variable closure.
- **C# 6.0** released with .NET 4.6 and VS2015 (July 2015). Implemented by Roslyn. Features: initializers for automatically implemented properties, using directives to import static members, exception filters, element initializers, `await` in `catch` and `finally` , extension `Add` methods in collection initializers.
- **C# 7.0** released with .NET 4.7 and VS2017 (March 2017) Major new features: tuples, ref locals and ref return, pattern matching (including pattern-based switch statements), inline `out` parameter declarations, local functions, binary literals, digit separators, and arbitrary async returns.
- **C# 7.1** released with VS2017 v15.3 (August 2017) New features: async main, tuple member name inference, default expression, pattern matching with generics.
- **C# 7.2** released with VS2017 v15.5 (November 2017) New features: private protected access modifier, Span<T>, aka interior pointer, aka stackonly struct, everything else.
- **C# 7.3** released with VS2017 v15.7 (May 2018). New features: enum, delegate and `unmanaged` generic type constraints. `ref` reassignment. Unsafe improvements: `stackalloc` initialization, unpinned indexed `fixed` buffers, custom `fixed` statements. Improved overloading resolution. Expression variables in initializers and queries. `==` and `!=` defined for tuples. Auto-properties' backing fields can now be targeted by attributes.
- **C# 8.0** currently in preview, a beta version available with VS2019 v16.0 (April 2019). Expected new features: Non-nullable reference-types, `IAsyncEnumerable<T>` support, Ranges, and default interface methods.

In response to the OP's question:

> What are the correct version numbers for C#? What came out when? Why can't I find any answers about C# 3.5?

There is no such thing as C# 3.5 - the cause of confusion here is that the C# 3.0 is present in .NET 3.5. The language and framework are versioned independently, however - as is the CLR, which is at version 2.0 for .NET 2.0 through 3.5, .NET 4 introducing CLR 4.0, service packs notwithstanding. The CLR in .NET 4.5 has various improvements, but the versioning is unclear: in some places it may

As of May 3, 2017, the C# Language Team created a history of C# versions and features on their github repo: Features Added in C# Language Versions. There is also a page that tracks upcoming and recently implemented language features.

edited Jul 14 at 16:56

community wiki
44 revs, 25 users 51%
Jon Skeet

15 To whoever suggested including concurrent collections: this is a list of *language* features, not *framework* features. Note the lack of mentioning WPF, etc. – Jon Skeet Mar 25 '14 at 11:52

2 @nawfal: Roslyn is irrelevant to that - and .NET native is somewhat separate. But basically, yes, I believe it's still 4. – Jon Skeet Aug 20 '15 at 9:53

2 @nawfal: None of the language changes need any CLR changes. – Jon Skeet Aug 20 '15 at 10:00

8 @alper: Unity wouldn't be a specific version of C# so much as a specific version of the .NET framework and/or runtime. IIRC, it's effectively on CLR v2, but may have some aspects of .NET 3.5. – Jon Skeet Oct 14 '15 at 15:26

2 @markmnl: A project doesn't generally *have* a specified C# version number... you could open the same project in different versions of Visual Studio and find the same code works in one but doesn't work in another. You *can* limit the C# version, although that's done on a syntactic rather than semantic basis. But yes, if you create a project targeting .NET 4 in Visual Studio 2015, you can use most C# 6 features... – Jon Skeet May 19 '16 at 5:36

288

The biggest problem when dealing with C#'s version numbers is the fact that it is not tied to a version of the .NET Framework, which it appears to be due to the synchronized releases between Visual Studio and the .NET Framework.

The version of C# is actually bound to the compiler, not the framework. For instance, in Visual Studio 2008 you can write C# 3.0 and target .NET Framework 2.0, 3.0 and 3.5. The C# 3.0 nomenclature describes the version of the code syntax and supported features in the same way that ANSI C89, C90, C99 describe the code syntax/features for C.

Take a look at Mono, and you will see that Mono 2.0 (mostly implemented version 2.0 of the .NET Framework from the ECMA specifications) supports the C# 3.0 syntax and features.

edited Nov 6 '17 at 17:19

community wiki
2 revs, 2 users 53%
Bryan Rehbein

285

C# version	VS version	.NET version	CLR version	Release date
1.0	2002	1.0	1.0	Feb 2002
1.2	2003	1.1	1.1	Apr 2003
2.0	2005	2.0	2.0	Nov 2005
		3.0	2.0	Nov 2006
3.0	2008	3.5	2.0	Nov 2007
4.0	2010	4.0	4	Apr 2010
5.0	2012	4.5	4	Aug 2012
5.0	2013	4.5.1	4	Oct 2013
		4.5.2	4	May 2014
6.0	2015	4.6	4	Jul 2015
		4.6.1	4	Nov 2015
		4.6.2	4	Aug 2016
7.0	2017			Mar 2017
		4.7	4	May 2017
7.1	2017(v15.3)			Aug 2017
		4.7.1	4	Oct 2017
7.2	2017(v15.5)			Dec 2017
		4.7.2	4	Apr 2018
7.3	2017(v15.7)			May 2018
8.0	2019			Apr 2019

Note: .NET development is pretty much independent of VS these days, there is no correlation between versions of each. Refer to ".NET Framework versions and dependencies" for more.

edited Apr 17 at 20:42

community wiki
12 revs, 5 users 92%
nawfal

5 What about .NET Core versions ? – Pac0 Jan 26 '18 at 12:14

1 @Pac0 .NET Core development is pretty much independent of VS development and is hard to keep track of (being open source). In fact all of VS, .NET and .NET Core are iterating quite fast. – nawfal Feb 6 '18 at 13:15

- **C# 1.0** with Visual Studio.NET

- **C# 4.0** with Visual Studio 2010
- **C# 5.0** with Visual Studio 2012
- **C# 6.0** with Visual Studio 2015
- **C# 7.0** with Visual Studio 2017
- **C# 8.0** with Visual Studio 2019 (in preview)

edited Aug 28 at 22:45

community wiki
6 revs, 6 users 58%
iTSrAVIE

76

VERSION ____ LANGUAGE SPECIFICATION _____ MICROSOFT COMPILER

C# 1.0/1.2 ___ December 2001[?]/2003[?] _________ January 2002[?]

C# 2.0 ______ September 2005 _____________ November 2005[?]

C# 3.0 ______ May 2006 _________________ November 2006[?]

C# 4.0 ______ March 2009 (draft) ___________ April 2010[?]

C# 5.0; released with .NET 4.5 in August 2012

C# 6.0; released with .NET 4.6 2015

C# 7.0; released with .NET 4.7 2017

edited May 25 '17 at 17:34

community wiki
5 revs, 5 users 43%
maxwellb

7 Where did you get a C# 2.0 language specification in December 2002 from? Likewise C# 4 in June 2006? Are you sure you're not talking about ECMA editions, which are completely different? – Jon Skeet May 7 '10 at 11:28

4 just refer the following link en.wikipedia.org/wiki/C_Sharp_(programming_language) – Pramodh May 7 '10 at 11:33

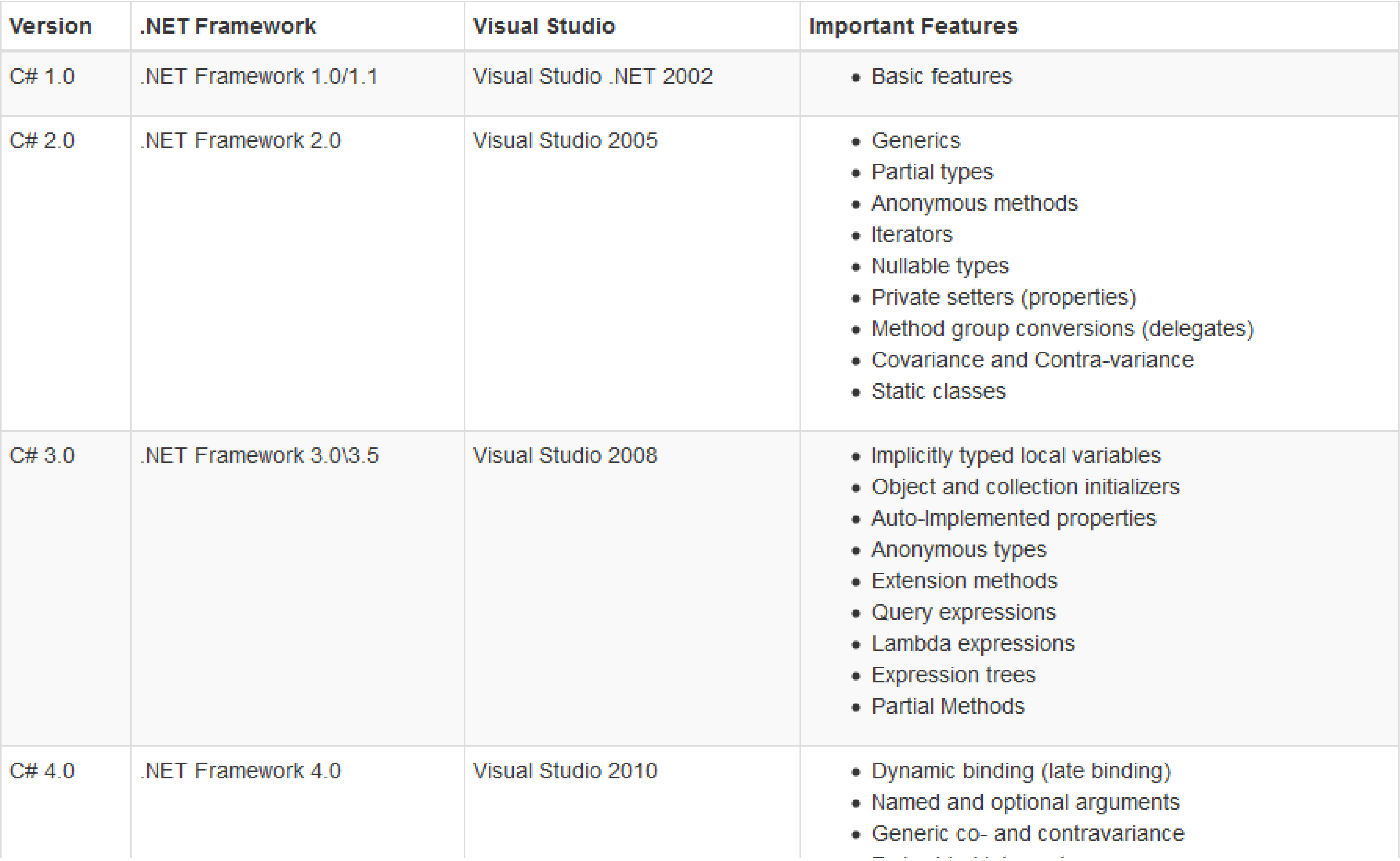
54

C# Version History:

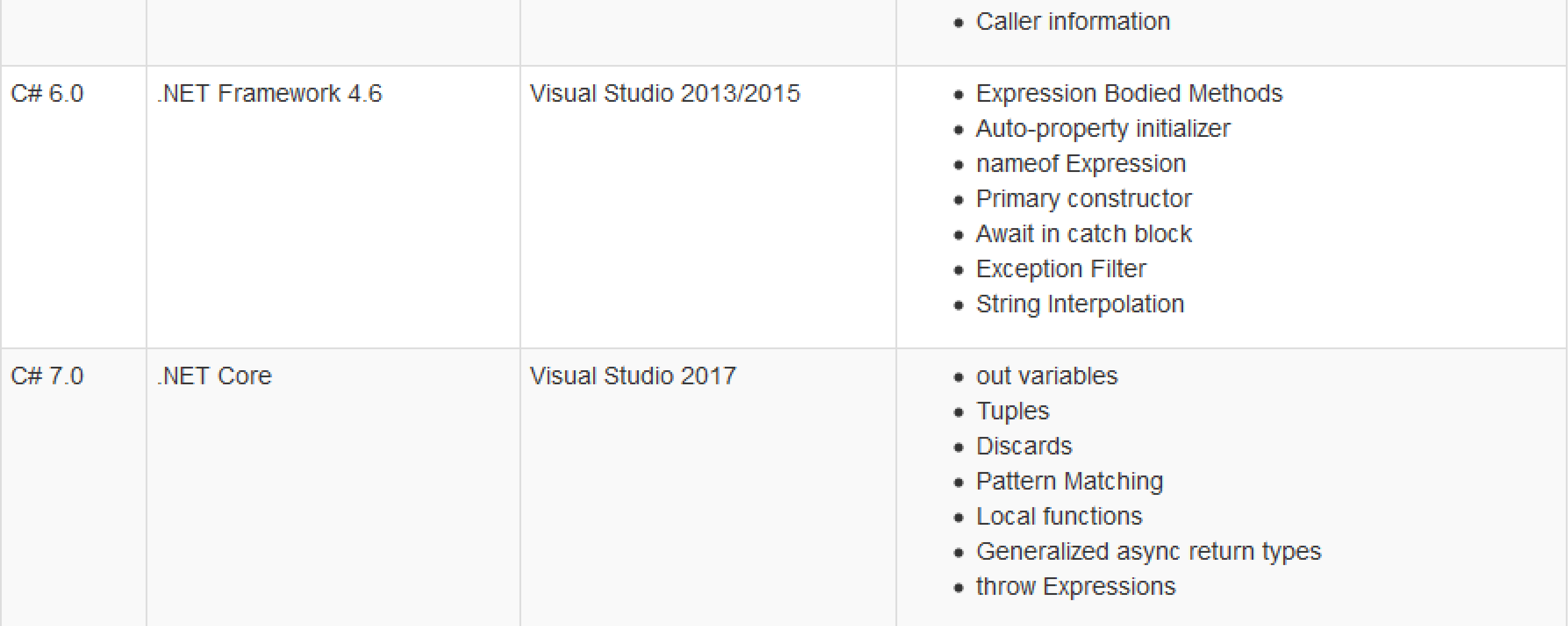
C# is a simple and powerful object-oriented programming language developed by Microsoft.

C# has evolved much since its first release in 2002. C# was introduced with .NET Framework 1.0.

The following table lists important features introduced in each version of C#.



Version	.NET Framework	Visual Studio	Important Features
C# 1.0	.NET Framework 1.0/1.1	Visual Studio .NET 2002	• Basic features
C# 2.0	.NET Framework 2.0	Visual Studio 2005	• Generics • Partial types • Anonymous methods • Iterators • Nullable types • Private setters (properties) • Method group conversions (delegates) • Covariance and Contra-variance • Static classes
C# 3.0	.NET Framework 3.0\3.5	Visual Studio 2008	• Implicitly typed local variables • Object and collection initializers • Auto-Implemented properties • Anonymous types • Extension methods • Query expressions • Lambda expressions • Expression trees • Partial Methods
C# 4.0	.NET Framework 4.0	Visual Studio 2010	• Dynamic binding (late binding) • Named and optional arguments • Generic co- and contravariance • Embedded interop types



			• Caller information
C# 6.0	.NET Framework 4.6	Visual Studio 2013/2015	• Expression Bodied Methods • Auto-property initializer • nameof Expression • Primary constructor • Await in catch block • Exception Filter • String Interpolation
C# 7.0	.NET Core	Visual Studio 2017	• out variables • Tuples • Discards • Pattern Matching • Local functions • Generalized async return types • throw Expressions

And the latest version of C# is available in [C# Versions](#).

edited Nov 6 '17 at 17:24

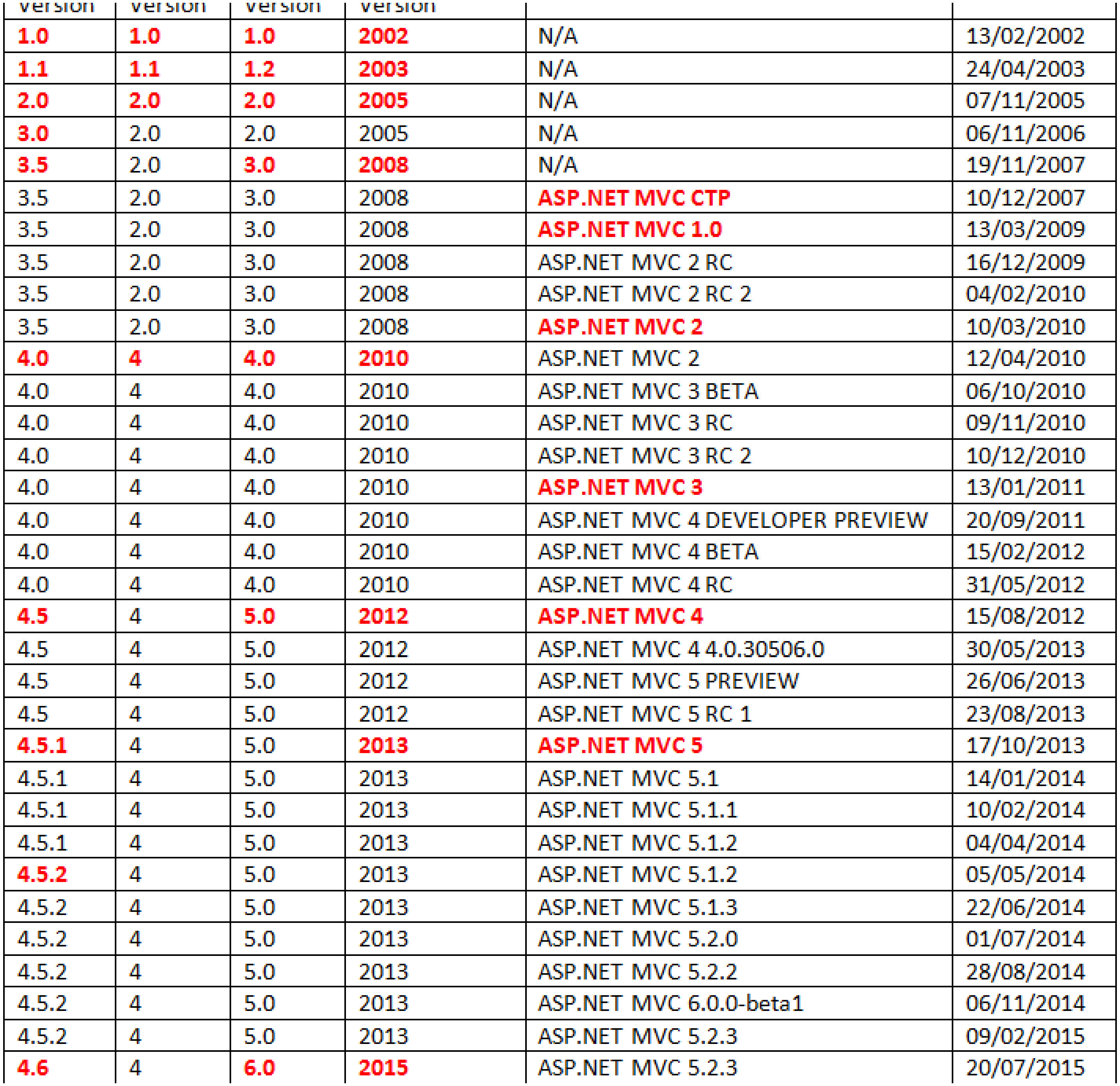
community wiki
6 revs, 4 users 61%
Milan

49

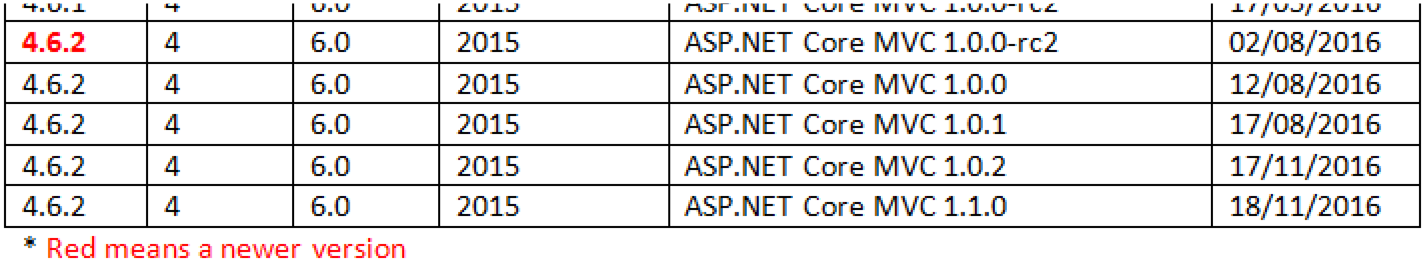
I've summarised most of the versions in this table. The only ones missing should be ASP.NET Core versions. I've also added different versions of ASP.NET MVC.

Note that ASP.NET 5 has been rebranded as ASP.NET Core 1.0 and ASP.NET MVC 6 has been rebranded as ASP.NET Core MVC 1.0.0. I believe this change occurred sometime around Jan 2016.

I have included the release date of ASP.NET 5 RC1 in the table, but I've yet to include ASP.NET core 1.0 and other core versions, because I couldn't find the exact release dates. You can read more about the release dates regarding ASP.NET Core here: [When is ASP.NET Core 1.0 (ASP.NET 5 / vNext) scheduled for release?](#)



version	version	version	version		
1.0	**1.0**	**1.0**	**2002**	N/A	13/02/2002
1.1	**1.1**	**1.2**	**2003**	N/A	24/04/2003
2.0	**2.0**	**2.0**	**2005**	N/A	07/11/2005
3.0	2.0	2.0	2005	N/A	06/11/2006
3.5	2.0	**3.0**	**2008**	N/A	19/11/2007
3.5	2.0	3.0	2008	**ASP.NET MVC CTP**	10/12/2007
3.5	2.0	3.0	2008	**ASP.NET MVC 1.0**	13/03/2009
3.5	2.0	3.0	2008	ASP.NET MVC 2 RC	16/12/2009
3.5	2.0	3.0	2008	ASP.NET MVC 2 RC 2	04/02/2010
3.5	2.0	3.0	2008	**ASP.NET MVC 2**	10/03/2010
4.0	**4**	**4.0**	**2010**	ASP.NET MVC 2	12/04/2010
4.0	4	4.0	2010	ASP.NET MVC 3 BETA	06/10/2010
4.0	4	4.0	2010	ASP.NET MVC 3 RC	09/11/2010
4.0	4	4.0	2010	ASP.NET MVC 3 RC 2	10/12/2010
4.0	4	4.0	2010	**ASP.NET MVC 3**	13/01/2011
4.0	4	4.0	2010	ASP.NET MVC 4 DEVELOPER PREVIEW	20/09/2011
4.0	4	4.0	2010	ASP.NET MVC 4 BETA	15/02/2012
4.0	4	4.0	2010	ASP.NET MVC 4 RC	31/05/2012
4.5	4	**5.0**	**2012**	**ASP.NET MVC 4**	15/08/2012
4.5	4	5.0	2012	ASP.NET MVC 4 4.0.30506.0	30/05/2013
4.5	4	5.0	2012	ASP.NET MVC 5 PREVIEW	26/06/2013
4.5	4	5.0	2012	ASP.NET MVC 5 RC 1	23/08/2013
4.5.1	4	5.0	**2013**	**ASP.NET MVC 5**	17/10/2013
4.5.1	4	5.0	2013	ASP.NET MVC 5.1	14/01/2014
4.5.1	4	5.0	2013	ASP.NET MVC 5.1.1	10/02/2014
4.5.1	4	5.0	2013	ASP.NET MVC 5.1.2	04/04/2014
4.5.2	4	5.0	2013	ASP.NET MVC 5.1.2	05/05/2014
4.5.2	4	5.0	2013	ASP.NET MVC 5.1.3	22/06/2014
4.5.2	4	5.0	2013	ASP.NET MVC 5.2.0	01/07/2014
4.5.2	4	5.0	2013	ASP.NET MVC 5.2.2	28/08/2014
4.5.2	4	5.0	2013	ASP.NET MVC 6.0.0-beta1	06/11/2014
4.5.2	4	5.0	2013	ASP.NET MVC 5.2.3	09/02/2015
4.6	4	**6.0**	**2015**	ASP.NET MVC 5.2.3	20/07/2015



4.6.1	4	6.0	2015	ASP.NET Core MVC 1.0.0-rc2	17/05/2016
4.6.2	4	6.0	2015	ASP.NET Core MVC 1.0.0-rc2	02/08/2016
4.6.2	4	6.0	2015	ASP.NET Core MVC 1.0.0	12/08/2016
4.6.2	4	6.0	2015	ASP.NET Core MVC 1.0.1	17/08/2016
4.6.2	4	6.0	2015	ASP.NET Core MVC 1.0.2	17/11/2016
4.6.2	4	6.0	2015	ASP.NET Core MVC 1.1.0	18/11/2016

* Red means a newer version

edited Nov 6 '17 at 17:31

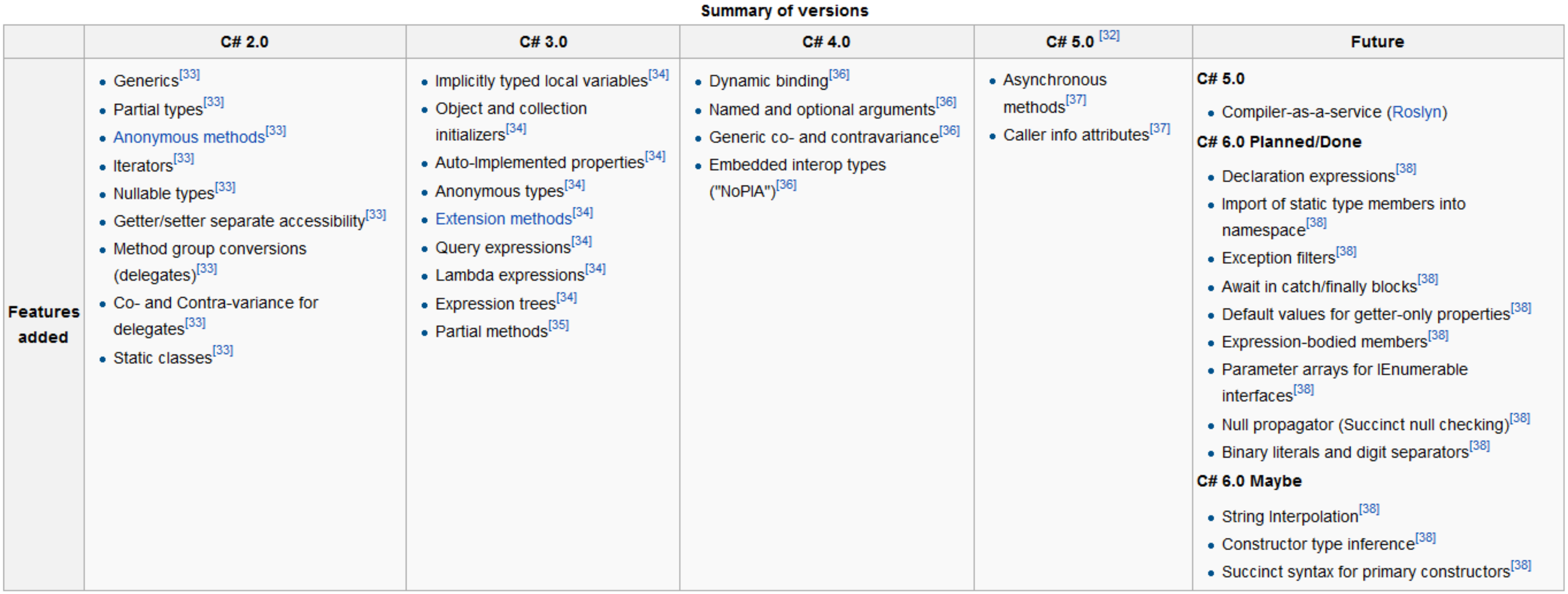
community wiki
5 revs, 2 users 83%
Mindless

1 I'm not sure that having MVC in the same table is helpful, to be honest... it's just on a separate release schedule, effectively. – Jon Skeet Jan 4 '17 at 7:05

@Jon This is true, just adding it here for anyone that might need it, because i did try to find out the correponding release dates of .NET frameworks, so that i get a better understanding of the whole version history. – Mindless Jan 4 '17 at 7:22

44

You can check the latest C# versions here



Summary of versions

	C# 2.0	C# 3.0	C# 4.0	C# 5.0 [32]	Future
Features added	• Generics[33] • Partial types[33] • Anonymous methods[33] • Iterators[33] • Nullable types[33] • Getter/setter separate accessibility[33] • Method group conversions (delegates)[33] • Co- and Contra-variance for delegates[33] • Static classes[33]	• Implicitly typed local variables[34] • Object and collection initializers[34] • Auto-Implemented properties[34] • Anonymous types[34] • Extension methods[34] • Query expressions[34] • Lambda expressions[34] • Expression trees[34] • Partial methods[35]	• Dynamic binding[36] • Named and optional arguments[36] • Generic co- and contravariance[36] • Embedded interop types ("NoPIA")[36]	• Asynchronous methods[37] • Caller info attributes[37]	**C# 5.0** • Compiler-as-a-service (Roslyn) **C# 6.0 Planned/Done** • Declaration expressions[38] • Import of static type members into namespace[38] • Exception filters[38] • Await in catch/finally blocks[38] • Default values for getter-only properties[38] • Expression-bodied members[38] • Parameter arrays for IEnumerable interfaces[38] • Null propagator (Succinct null checking)[38] • Binary literals and digit separators[38] **C# 6.0 Maybe** • String Interpolation[38] • Constructor type inference[38] • Succinct syntax for primary constructors[38]

edited Oct 30 '14 at 19:58

community wiki
3 revs, 2 users 91%
Israel Ocbina

32

Comparing the MSDN articles "What's New in the C# 2.0 Language and Compiler" and "What's New in Visual C# 2005", it is possible to deduce that "C# major_version.minor_version" is coined according to the compiler's version numbering.

There is C# 1.2 corresponding to .NET 1.1 and VS 2003 and also named as Visual C# .NET 2003.

But further on Microsoft stopped to increment the minor version (after the dot) numbers or to have them other than zero, `0` . Though it should be noted that C# corresponding to .NET 3.5 is named in msdn.microsoft.com as "Visual C# 2008 Service Pack 1".

There are two parallel namings: By major .NET/compiler version numbering and by Visual Studio numbering.

C# 2.0 is a synonym for Visual C# 2005

- .NET3.0 <==> Visual C# 2008
- .NET 3.5 <==> Visual C# 2008 Service Pack 1

edited Nov 6 '17 at 17:22

community wiki
2 revs, 2 users 81%
Gennady Vanin Геннадий Ванин

3 No, C# corresponding to .NET 3.5 is named "Visual C# 2008" if you really want to use that numbering. The C# 3.0 features were introduced in "Visual C# 2008" which is why on the page you're linked to they're under "What's New in the Original Release Version of Visual C# 2008". Using the Visual Studio version numbers is a bad idea in general though, as it makes very little sense when you're building with Mono, for example. The C# language has well-specified version numbers... we know which Visual C# product originally introduced that version of C#, but they're not the same thing. – Jon Skeet May 21 '13 at 10:22

@JonSkeet, no, I don't. Wanted to ask you (and another answerer) update your answer but since my comment became too lengthy, I've decided then to put as answer. Thanks for your info – Gennady Vanin Геннадий Ванин May 21 '13 at 13:32 ✎

I dont think C# 3.0 can run on VS 2005. – nawfal Jul 21 '16 at 14:22

protected by user7116 Nov 5 '12 at 16:16

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?