

# Guid.NewGuid() vs. new Guid()

▲ What's the difference between `Guid.NewGuid()` and `new Guid()` ?

313 Which one is preferred?

▼ c# guid



33

edited Aug 13 '12 at 16:25



josh3736

108k 22 181 232

asked Aug 13 '12 at 16:10



OscarRyz

147k 99 345 521

6 @ClintonWard I'm practically new to C# and @ Bob2Chiv I can't run the project right now and was curious about it. – OscarRyz Aug 13 '12 at 16:21

6 @OscarRyz - For quickly testing code in C#, I use [LinqPad](#). – DaveShaw Aug 13 '12 at 20:35

5 Actually I do believe this question is relevant, because it's confusing and I don't really "00000000-0000-0000-0000-000000000000" as the best default. I just ran into a problem where a client couldn't login to a system because somewhere deep inside the code it was using that GUID. – Michiel Cornille Mar 14 '13 at 10:05

2 @DaveShaw - LinqPad is great and I use it a lot. But I wanted to point out that these kinds of tests. – Mark Meuer Aug 24 '16 at 16:06

1 @MichielCornille Just got to know the feeling bro... – Jins Peter Jan 9 at 14:45



OscarRyz

147k 99 345 521  
Mexico

I'm a 0x28 years old software developer who happens to like writing code.

Here are some interesting answers you might like to upvote :")

Why java people frequently consume exception silently ?  
Codi...

How to create a... History of Obje...

## 4 Answers

▲ `new Guid()` makes an "empty" all-0 guid (00000000-0000-0000-0000-000000000000 is not very useful).

523 `Guid.NewGuid()` makes an actual guid with a unique value, what you probably want.

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



- 24 An empty UUID is very useful indeed. It makes a great special value. – [Jon Hanna](#) Aug 13 '12 at 16:21
- 89 On a side note, `new Guid()` is equivalent to `Guid.Empty`. – [Steve Guidi](#) Aug 13 '12 at 16:24
- 29 @JonHanna All guids make great special values. Unfortunately, the empty ones have a tendency to collide. I agree that empty guids are useful though, typically to indicate that something is uninitialized. – [MarkPflug](#) Aug 13 '12 at 16:27
- 5 The empty ones are very useful because they do collide. A special value that didn't collide with the known special value would be useless. – [Jon Hanna](#) Aug 13 '12 at 16:44
- 5 I think you're both agreeing that it makes a good "known guid" to compare to, as in the case of indicating something is awaiting initialization or is in some other known state. Personally I'd prefer to use a null value, but I can see that somebody might need a "special" guid at some point, and the all-0 guid is probably the guid least likely to violate the principal of least surprise for future maintainers of the code. – [PeterL](#) Aug 13 '12 at 19:32



`Guid.NewGuid()` creates a new UUID using an algorithm that is designed to make collisions very, very unlikely.

37

`new Guid()` creates a UUID that is all-zeros.



Generally you would prefer the former, because that's the point of a UUID (unless you're receiving it from somewhere else of course).

There are cases where you do indeed want an all-zero UUID, but in this case `Guid.Empty` or `default(Guid)` is clearer about your intent, and there's less chance of someone reading it expecting a unique value had been created.

In all, `new Guid()` isn't that useful due to this lack of clarity, but it's not possible to have a value-type that doesn't have a parameterless constructor that returns an all-zeros-and-nulls value.

Edit: Actually, it is possible to have a parameterless constructor on a value type that doesn't set everything to zero and null, but you can't do it in C#, and the rules about when it will be called and when there will just be an all-zero struct created are confusing, so it's not a good idea anyway.

edited Jan 29 '14 at 9:57

answered Aug 13 '12 at 16:19



[Jon Hanna](#)

93.5k 9 117 217

- 14 I'll add this for fun. For a 1% chance of collision, you'd need to generate about 2,600,000,000,000,000 GUIDs – [Clinton Ward](#) Aug 13 '12 at 16:20

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



that's deliberately messing with the input. [Jon Hanna](#) Aug 13 '12 at 16:29

- 2 That was v1 Guid. newer MS GUIDs are V4 and do not use the MAC address as part of the GUID generation. Time is still a factor though – [Clinton Ward](#) Aug 13 '12 at 16:34



16



**Guid.NewGuid()**

Eventually calls CoCreateGuid (a COM call to Ole32) (reference [here](#)) and the actual work is done by [UuidCreate](#).

Guid.Empty is meant to be used to check if a Guid contains all zeroes. This could also be done via comparing the value of the Guid in question with **new Guid()**

So, if you need a **unique identifier**, the answer is **Guid.NewGuid()**

answered Apr 8 '14 at 6:00



[Sudhanshu Mishra](#)

4,164 2 39 57



-2



Guid.NewGuid() , as it creates GUIDs as intended.

Guid.NewGuid() creates an empty Guid object, initializes it by calling CoCreateGuid and returns the object.

new Guid() merely creates an empty GUID (all zeros, I think).

I guess they had to make the constructor public as Guid is a struct .

edited Apr 23 '18 at 6:34



[Vadim Ovchinnikov](#)

7,821 4 29 53

answered Aug 31 '16 at 3:48



[Sharath](#)

15

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



---

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook

