

Re: Could an object refuse to be created!?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-05/3391.html>

From: Justin Rogers (*Justin_at_games4dotnet.com*)

Date: 05/16/04

Date: Sun, 16 May 2004 02:05:20 -0700

Would be nice to note, that in cases where your object creation is linked to valid input criterion, that you might instead use a factory method. Factory methods demonstrate, better than constructors, that work is going to occur and that validation of parameters might be necessary. Factory methods are also capable of returning null rather than throwing an exception. To then prevent the user from instantiating your type directly, you simply privatize the constructor.

```
--
Justin Rogers
DigiTec Web Consultants, LLC.
Blog: http://weblogs.asp.net/justin\_rogers
"Sunny" <sunnyask@icebergwireless.com> wrote in message
news:eJsk8GwOEHA.644@tk2msftngpl3.phx.gbl...
> Hi Kevin, I know what you mean. My idea was that OP has to read the
> documentation :)
>
> During all these years thats the only think I have learned :)
> And of course, I have started with: "If nothing else helps, read the
> docs" :)
>
> Sunny
>
> Kevin P. Fleming wrote:
>
> > Sunny wrote:
> >> Kevin P. Fleming wrote:
> >>
> >>>Yes, throw an Exception in the object's constructor. This can be
> >>>somewhat ugly to deal with depending on how the object's user is trying
> >>>to instantiate the object, but it will definitely keep the object from
> >>>being created.
> >>
> >>
> >> Actually, it is not ugly. This approach is used all over the .net.
> >
> > Oh, I know. I was trying to make sure the OP knew that things like this:
> >
> > private void Foo()
> > {
> >     SomeObjectType obj = new SomeObjectType();
> > }
> >
```

microsoft.public.dotnet.languages.csharp: Re: Could an object refuse to be created!?

> > can be a pain to deal with if SomeObjectType's constructor throws an
> > exception. It's even worse if these initializations are at the type
> > level, and if they are for static fields, then they could happen any
> > time at all and be *_very_* confusing.
> >
> > That doesn't it's not the right way to do it though; given a choice
> > between the constructor throwing an exception and having the object be
> > in "limbo" until it's initialized I'd take the former every time :-)
>