

## Re: Opposite of typeof /GetType()

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2008-03/msg01120.html>

---

- *From:* "pipo" <nomail@xxxxxx>
  - *Date:* Mon, 10 Mar 2008 17:03:46 +0100
- 

Sorry, if I misunderstood your question.

```
public class Foo<T>
where T : new()
{
protected T bar()
{
return new T();
}
}
public class SuperFoo : Foo<SuperFoo>
{
//Class code
}
```

```
SuperFoo super = new Foo<SuperFoo>().bar();
```

"Ade" <madge@xxxxxxxxxxxx> schreef in bericht  
[news:b0046e01-9c4e-4cba-8bb6-5eb02cf9dd16@xx](mailto:news:b0046e01-9c4e-4cba-8bb6-5eb02cf9dd16@xx)

Whereas typeof gets a Type from a class, I need to get a class from a Type so that I may create the type at runtime.

Imagine if you wil

```
class foo
{
public foo bar()
{
foo newFoo = new GetType();
// Some magic goes here
return newFoo;
}
}
```

What would be the syntactically-correct way to do this? I want to create, from the base class, a new instance of whatever type the

Re: Opposite of typeof /GetType()

instance of foo is. eg calling bar() on

class superfoo : foo

will give me something unboxable to a superfoo.