

We now integrate with Microsoft Teams, helping you to connect your internal knowledge base with your chat. [Learn more.](#)

# What is a read only collection?

Asked 9 years, 6 months ago   Active 3 years, 11 months ago   Viewed 21k times



20



3

I ran a security code analyst i found myself having a [CA2105 warning](#). I looked at the grade tampering example. I didn't realize you can assign `int[]` to a `readonly int`. I thought `readonly` was like the C++ `const` and makes it illegal.

The How to Fix Violations suggest i clone the object (which i don't want to do) or 'Replace the array with a strongly typed collection that cannot be changed'. I clicked the link and see 'ArrayList' and adding each element one by one and it doesn't look like you can prevent something adding more.

So when i have this piece of code what is the easiest or best way to make it a read only collection?

```
public static readonly string[] example = { "a", "b", "sfsdg", "sdgfhf", "erfdgf", "last  
one"};
```

c#

wcf

readonly

readonly-collection

edited Apr 21 '10 at 6:52



John Saunders

150k 23 212 374

asked Apr 21 '10 at 6:43

user34537

possible duplicate of [Return collection as read-only](#) – nawfal Nov 11 '13 at 11:01

A `readOnlyCollection` = Normal collection - Add/remove APIs - Set Indexer – RBT Feb 20 '17 at 1:56

## 6 Answers



The easiest way to have a collection which you cannot modify is to use

36

[ReadOnlyCollection](#)

Example from MSDN:



```
List<string> dinosaurs = new List<string>();
dinosaurs.Add("Tyrannosaurus");
dinosaurs.Add("Amargasaurus");
dinosaurs.Add("Deinonychus");
dinosaurs.Add("Compsognathus");
```

```
ReadOnlyCollection<string> readOnlyDinosaurs = new ReadOnlyCollection<string>
(dinosaurs);
```

answered Apr 21 '10 at 6:48

**Mikael Svenson****33.5k** 4 65 69

11

```
public static readonly ReadOnlyCollection<string> example
    = new ReadOnlyCollection<string>(new string[] { "your", "options", "here" });
```

(although it should still probably be exposed as a `get` property rather than a public field)

answered Apr 21 '10 at 6:46

**Marc Gravell** ♦**831k** 216 2238  
2636

1 I would second here on using the `readonly` keyword for the variable as well. Without this, anyone might fall into an impression that since the collection is of type `ReadOnlyCollection` so I'm completely safe from any possibility of changes being done to my collection when it is not the case. One might not realize that the collection is read-only but the variable pointing to it is not. What a `ReadOnlyCollection` would mean if I can reassign the variable itself to a new collection altogether. +1. – [RBT](#) Feb 17 '17 at 1:44

1 @RBT note that `readonly` is a lie; `readonly` fields can absolutely be mutated. You have to cheat a little, of course... – [Marc Gravell](#) ♦ Feb 17 '17 at 2:01

It took me a while to digest [this](#). Thank you for showing me the light. – [RBT](#) Feb 22 '17 at 23:56



If you're working with arrays, you can use

8

```
return Array.AsReadOnly(example);
```



to wrap your array in a read-only collection.

answered Apr 21 '10 at 13:35



[Steve Dennis](#)

1,283 10 9



```
var readOnly = new ReadOnlyCollection<string>(example);
```

3



answered Apr 21 '10 at 6:46



[Mark Seemann](#)

191k 34 345 601



```
ReadOnlyCollection<string> readOnlyCollection =  
    new ReadOnlyCollection<string>(example);
```

1



answered Apr 21 '10 at 6:47



[Kyle Rozendo](#)

22.1k 6 70 113



I was looking for a similar solution, but I wanted to still be able to modify the collection from inside the class so I chose the option outlined here: <http://www.csharp-examples.net/readonly-collection/>

1

In short, his example is:



```
public class MyClass  
{  
    private List<int> _items = new List<int>();
```

```
public IList<int> Items
{
    get { return _items.AsReadOnly(); }
}
```

edited Dec 7 '15 at 14:18

**TheLethalCoder****5,610** 6 26 49

answered Feb 21 '13 at 0:48

**nelsonmichael****74** 1 3

---