

## How do I automatically add or remove methods from all implementations of an interface in VS2010?

[Ask Question](#)

- ▲  
3  
▼  
★  
2
- In my current project, using Visual Studio 2010 and C#, I often find myself having to update interfaces that I've defined to add new or remove old methods and members of the interface. As a result, I have to look back through my codebase, find all the classes that implement the interface, and manually remove all occurrences of those obsolete methods and members, or add new ones in by selecting "Implement Interface" for each class. This is highly tedious, and it's slowing down my workflow.

Is there any way to automatically add placeholder implementations of new methods or members for interfaces to all the classes that implement the interface, without having to go through each class? Similarly, is there any way to remove a method or member from a class at the same time as I remove it from the interface the class implements?

[c#](#)[visual-studio-2010](#)[shortcut](#)

asked Dec 19 '12 at 6:14



[akbiggs](#)

11.6k 5 18 34

I do not think so. But the benefit you get from the current VS behaviour is that you will get compile time error when you add a method to your interface, which is really great. So you know you have to implement them in your derived classes too. While removing may not benefit from this, it doesn't give you any problem too since having a redundant method in sub classes is ok for the compiler. Now I think Resharper can tell you if you have redundant methods and you can easily delete them.. – [nawfal](#) Dec 19 '12 at 6:32

[Home](#)[PUBLIC](#)[Stack Overflow](#)[Tags](#)[Users](#)[Jobs](#)**Teams**

Q&amp;A for work

[Learn More](#)

## 2 Answers

▲  
5  
▼



If you are using [ReSharper](#) you can just add the method to the interface, click `Alt+Enter` and choose 'Add method to derived classes':

```
public interface IMyAwesomeInterface
{
    void DoSomethingAwesome();
}

public class MyAwesomeClass : IMyAwesomeInterface
{
}
```

If you are removing a method, you can use `Refactor > Safe Delete ...` functionality:

```
public interface IMyAwesomeInterface
{
    void DoSomethingAwesome();
}

public class MyAwesomeClass : IMyAwesomeInterface
{
    public void DoSomethingAwesome()
    {
        throw new NotImplementedException();
    }
}
```

answered Dec 19 '12 at 10:09



RoelF

5,850

3

35

63

---

Awesome. Just started using ReSharper...this tool is great. – [akbiggs](#)

Dec 21 '12 at 20:05

---

Telerik JustCode have the same option as 'Add stub to all Inheritors' and 'safe Delete' – [voddy](#) Apr 3 '14 at 0:58

---

Short answer: no.

3

One possibility would be: not to update the interface, but add an *extension method* on the interface, for example:

```
public static class SomeUtilityClass
{
    public static void DoTheAwesomez(this ISomeInterface obj, int count)
    {
        for(int i = 0 ; i < count ; i++)
        {
            obj.Foo();
            obj.Bar();
        }
    }
}
```

then if you have:

```
ISomeInterface x = ...
x.DoTheAwesomez(20);
```

But for a regular interface method: no, there's no automatic way of adding it, but simply hit "build", and the compiler will tell you which types are missing it. Adding the interface implementation to a

common base class will often make it easy to hit multiple types at once.

For *removing* the method: one way to do that is to use *explicit interface implementation*:

```
class Foo : ISomeInterface {  
    void ISomeInterface.SomeMethod(string x) {  
        ....  
    }  
}
```

Now, if you remove the `SomeMethod` method from `ISomeInterface`, the compiler *will tell you* about all the places it is used.

Other low-tech ways of finding all the `SomeMethod` methods include:

- `ctrl` `shift` `f` (global find) - low tech, but works...
- just before you delete the method from the interface, rename it **without using refactoring** to `SomeMethod_xyz` (or whatever): now, all the types that implement this will complain that they don't implement `SomeMethod_xyz`, and all the types that *consume* this will complain that `SomeMethod` doesn't exist

answered Dec 19 '12 at 6:59



Marc Gravell ♦

791k 195 2156

2559

