Interfaces and Versioning

Asked 16 years, 3 months ago Modified 12 years, 8 months ago

Viewed 3k times

12

I am designing a new System and I have a lot of Interfaces that will grow over time with the system. What is the best practice to name this interfaces

```
ISomethingV01
ISomethingV02
etc
```

and I do this

```
public interface ISomething{
      void method();
}
```

then I have to add method 2 so now what I do?

```
public interface ISomethingV2:ISomething{
      void method2();
}
```

or same other way?

naming-conventions interface versioning

Share

Improve this question

Follow

edited Jun 20, 2009 at 16:39

Quinn Taylor

44.8k ● 16 ● 115 ● 133

asked Sep 5, 2008 at 2:15

Jedi Master Spooky

5,809 ● 14 ● 59 ● 86

5 Answers

Sorted by: Highest score (default)

7

I think you're overrusing interfaces.

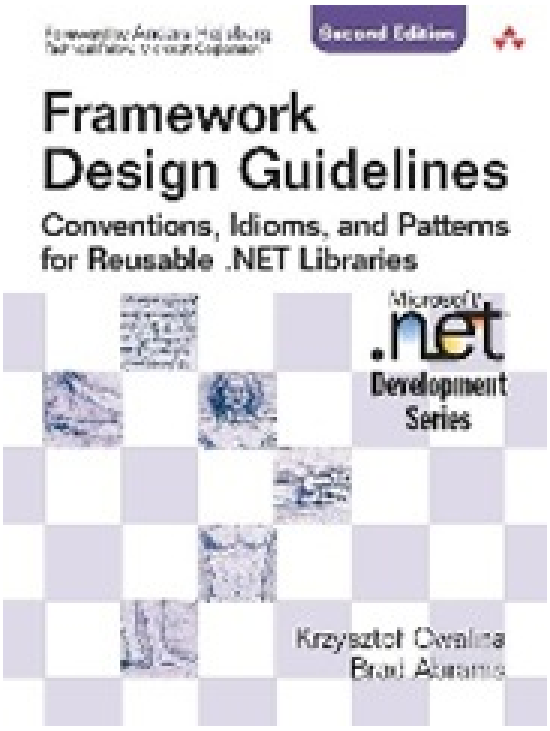
Meyer and Martin told us: "Open for extension but closed for modification!"

and then Cwalina (et al) reiterated:

From Framework Design Guidelines...

> In general, classes are the preferred construct for exposing abstractions. The main drawback of interfaces is that they are much less flexible than classes when it comes to allowing for evolution of APIs. Once you ship an interface, the set of its members is fixed forever. Any additions to the interface would break existing types implementing the interface.
>
> A class offers much more flexibility. You can add members to classes that have already shipped. As long as the method is not abstract (i.e., as

> long as you provide a default implementation of the method), any existing derived classes continue to function unchanged.



Share Improve this answer

Follow

edited Apr 2, 2012 at 8:37

Neysor

3,911 ● 11 ● 35 ● 66

answered Sep 5, 2008 at 2:47

rp.

17.6k ● 14 ● 65 ● 79

I am sorry. Could you elaborate more details with examples, why the method is not abstract? If it is abstract, what will happen? I am a newbie in programming. Thank you.
– Display Name Feb 18, 2019 at 7:23

Ideally, you shouldn't be changing your interfaces very often (if at all). If you do need to change an interface, you

5

should reconsider its purpose and see if the original name still applies to it.

If you still feel that the interfaces will change, and the interfaces changes are small (adding items) and you have control of the whole code base, then you should just modify the interface and fix all the compilation errors.

If your change is a change in how the interface is to be used, then you need to create a separate interface (most likely with a different name) to support that alternative usage pattern.

Even if you end up creating ISomething, ISomething2 and ISomething3, the consumers of your interfaces will have a hard time figuring out what the differences are between the interfaces. When should they use ISomething2 and when should they use ISomething3? Then you have to go about the process of obsoleting ISomething and ISomething2.

Share Improve this answer

Follow

answered Sep 5, 2008 at 2:29

Garo Yeriazarian

2,533 ● 18 ● 29

4

I agree with Garo Yeriazarian, changing interface is a serious decision. Also, if you want to promote usage of new version of interface you should mark old version as obsolete. In .NET you can add ObsoleteAttribute.

Share Improve this answer

edited May 23, 2017 at 12:13

Follow

Community Bot
1 • 1

answered Sep 5, 2008 at 2:39

aku
124k • 33 • 176 • 203

2

The purpose of an interface is to define an abstract pattern that at type must implement.

It would be better implement as:

```
public interface ISomething

public class Something1 : ISomething
public class Something2 : ISomething
```

You do not gain anything in the form of code reusability or scalable design by creating multiple versions of the same interface.

Share Improve this answer

Follow

answered Sep 5, 2008 at 2:18

FlySwat
175k • 75 • 248 • 314

2

I don't know why people downvote your post. I think that good naming guidelines are **very** important.

If you need to maintain compatibility with prev. version of the same interface consider using inheritance. If you

need to introduce new version of interface consider following rule:

> Try to add meaningful suffix to you interface. If it's not possible to create concise name, consider adding version number.

Share Improve this answer

Follow

answered Sep 5, 2008 at 2:20

aku

124k ● 33 ● 176 ● 203