Developer Tips: C# Enum Alternative

By Andrew Hinkle

Enum's are great for storing constants, but they lack being strongly typed because you can't create an instance of them. I want the best of both worlds. I want my switch statements to work off constants like enums. I want to pass these constants in a strongly typed container.

While [switch statements are often seen as a code smell](https://www.danylkoweb.com/Blog/real-world-refactoring-switch-statement-to-a-class-hierarchy-9G) there are still plenty of situations where they make since. Other Enum alternatives have been posted by [Steve Smith (Ardalis)](https://ardalis.com/enum-alternatives-in-c) and [Jimmy Bogard (Los Techies)](https://lostechies.com/jimmybogard/2008/08/12/enumeration-classes/) that are solving a similar solution and there are similarities here.

As an example, I'm working on a project where I want to pass around the ServiceLifetime in a strongly typed container. I created the ServiceLifetimeWrapper with the same constants plus NotSet. I always add a NotSet to my enum's to avoid invalid values parsing to the first enum value. I carry that mentality with me into the wrapper class, but as you'll see I don't really have that issue here when I parse back the instance using the FromName method.

To accomplish this enum/class combo I created a backing field to store a list of the available instances of the class as a singleton. It uses reflection magic to create the list based on the public constants within the class, but only once. The FromName method returns the SingleOrDefault and throws an exception if the name is not in the list.

I wanted to move this logic into a base class or helper class however I wanted to keep the constructor private or at least protected. I tried changing the "new" statement to Activator.CreateInstance and added the first parameter as an object array, but it still could not find the constructor. If anyone wants to take a turn at trying to abstract this method as a generic to a base class, post your solution and let me know. Thanks!

This enum alternative is fairly lightweight and works with switch statements. The main difference being that instead of storing your actual value in a string variable you pass around and use an instance of the class and access the name via the Name property. I've added plenty of unit tests to show usage.

Use the enum alternative class when you are:

* Only interested in the name
* You want to take advantage of switch statements
* You want to keep the content in a strongly typed container

Following [Singleton](https://csharpindepth.com/articles/singleton) techniques, this class is thread-safe and tested in a Parallel.For test.

The code base can be found in full along with another example in my github tips: <https://github.com/penblade/Tips/tree/master/Tips.EnumAlternative>.

# ServiceLifetimeWrapper



# ServiceLifetimeWrapperTest





## Conclusion

The code base can be found in full along with another example in my github tips: <https://github.com/penblade/Tips/tree/master/Tips.EnumAlternative>.

Are you frustrated with the Enum's lack of class features? Do you have an alternative implementation that meets the same objectives? Do you have any suggestions on how to improve this solution?