optional array Parameter in C# [duplicate]

Asked 7 years, 11 months ago Active 7 years, 11 months ago Viewed 22k times

25

1

> **Possible Duplicate:**
> passing an empty array as default value of optional parameter in c#

I am having a method that looks like below. Currently parameter `tags` is NOT optional

```
void MyMethod(string[] tags=null)
{
   tags= tags ?? new string[0];
   /*More codes*/
}
```

I want to make parameter `tags` as optional, as per `c#` , to make a parameter optional you can set a default value in method signature. I tried below hacks but none worked.

Code that didn't worked - 1

```
void MyMethod(string[] tags=new string[0]){}
```

Code that didn't worked - 2

```
void MyMethod(string[] tags={}){}
```

Please suggest me what i am missing.

I have already seen this question

Passing an empty array as default value of an optional parameter

c# c#-4.0

edited May 23 '17 at 11:54
Community ♦
1 1

asked Nov 21 '11 at 14:54
Rusi Nova
1,187 5 23 43

marked as duplicate by BoltClock ♦, sehe, Raymond Chen, Richard Harrison, Andrew Barber Nov 21 '11 at 19:39

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

Take a look here: stackoverflow.com/questions/3480382/… – nowaq Nov 21 '11 at 14:58

2 in this particular case, you might get away with C#3.0 compatible `void MyMethod(params string[] tags)` – sehe Nov 21 '11 at 14:58

1 What's wrong with `tags = tags ?? new string[0];` anyway? And if you have already seen the question, the accepted answer clearly states that you *can't* set a default value with the C# 4 syntax, and that you can use the `??` operator instead as you are doing. – BoltClock ♦ Nov 21 '11 at 14:58

2 @Rusi Nova: It is optional if you set a default value of `null` . – BoltClock ♦ Nov 21 '11 at 15:02

4 @RusiNova, we might be using a different definition of optional. To be clear, as far as the language is concerned, `void Foo(string[] tags = null)` indeed makes supplying the argument *optional* for the caller. Whether or not the null is *usable* inside your method is a different matter. – Anthony Pegram Nov 21 '11 at 15:03

2 Answers

44 ✓

The documentation for optional arguments says:

> A default value must be one of the following types of expressions:
>
> - a constant expression;
> - an expression of the form `new ValType()` , where `ValType` is a value type, such as an `enum` or a `struct` ;
> - an expression of the form `default(ValType)` , where `ValType` is a value type.

Since `new string[0]` is neither a constant expression nor a `new` statement followed by a value type, it cannot be used as a default argument value.

The first code excerpt in your question is indeed a good workaround:

```
void MyMethod(string[] tags = null)
{
   tags = tags ?? new string[0];
   // Now do something with 'tags'...
}
```

answered Nov 21 '11 at 14:58

Frédéric Hamidi

219k 32 424 439

+1 And the workaround is suggested in the accepted answer to the duplicate question. – BoltClock ♦ Nov 21 '11 at 15:00

1 This is correct but if you use MyMethod(params Object[] parameterName) You can omit the params array and the compiler will pass an empty array for you. MyMethod() works just fine. – jwize Feb 15 '14 at 9:08

Upvoted for linking and in-lining the relevant authoritative documentation. "Teach a man to fish..." – aaronsnoswell Oct 19 '17 at 23:44

3

not sure if i am getting it right this works.

```
 static void Main(string[] args)
       {
               TestMe();

       }


private static void TestMe(string[] param = null)
    {
        if (param == null)
        {
            Console.WriteLine("its empty");
        }
    }
```

also the value of param has to be compile time constant

answered Nov 21 '11 at 15:02

np-hard
3,325 4 43 71