

DIFFERENCE BETWEEN REF AND OUT PARAMETERS

Ref and out parameters are used to pass an argument within a method. In this article, you will learn the differences between these two parameters.

Ref

The ref keyword is used to pass an argument as a reference. This means that when value of that parameter is changed in the method, it gets reflected in the calling method. An argument that is passed using a ref keyword must be initialized in the calling method before it is passed to the called method.

Out

The out keyword is also used to pass an argument like ref keyword, but the argument can be passed without assigning any value to it. An argument that is passed using an out keyword must be initialized in the called method before it returns back to calling method.

Program with ref and out keyword

```
1. public class Example
2. {
3.     public static void Main() //calling method
4.     {
5.         int val1 = 0; //must be initialized
6.         int val2; //optional
7.
8.         Example1(ref val1);
9.         Console.WriteLine(val1); // val1=1
10.
11.        Example2(out val2);
12.        Console.WriteLine(val2); // val2=2
13.    }
14.
15.    static void Example1(ref int value) //called method
16.    {
```

```
17. value = 1;
18. }
19. static void Example2(out int value) //called method
20. {
21. value = 2; //must be initialized
22. }
23. }
24.
25. /* Output
26. 1
27. 2
28. */
```

Note

1. Do not be confused with the concept of passing by reference and the concept of reference type. These two concepts are not the same.
2. A value type or a reference type can be passed to method parameter by using ref keyword. There is no boxing of a value type when it is passed by reference.
3. Properties cannot be passed to ref or out parameters since internally they are functions and not members/variables.

Ref and out in method overloading

Both ref and out cannot be used in method overloading simultaneously. However, ref and out are treated differently at run-time but they are treated same at compile time (CLR doesn't differentiate between the two while it created IL for ref and out). Hence methods cannot be overloaded when one method takes a ref parameter and other method takes an out parameter. The following two methods are identical in terms of compilation.

```
1. class MyClass
2. {
3. public void Method(out int a) // compiler error "cannot define
   overloaded"
4. {
5. // method that differ only on ref and out"
6. }
```

```
7. public void Method(ref int a)
8. {
9. // method that differ only on ref and out"
10. }
11. }
```

However, method overloading can be done, if one method takes a ref or out argument and the other method takes simple argument. The following example is perfectly valid to be overloaded.

```
1. class MyClass
2. {
3. public void Method(int a)
4. {
5.
6. }
7. public void Method(out int a)
8. {
9. // method differ in signature.
10. }
11. }
```

Source : <http://www.dotnet-tricks.com/Tutorial/csharp/K0Hb060414-Difference-between-ref-and-out-parameters.html>