Ref Vs Out Keywords in C#

Introduction

The keywords ref and out are used to pass arguments within a method or function. Both indicate that an argument / parameter is passed by reference. By default parameters are passed to a method by value. By using these keywords (ref and out) we can pass a parameter by reference.

Ref Keyword

The ref keyword passes arguments by reference. It means any changes made to this argument in the method will be reflected in that variable when control returns to the calling method.

Example code

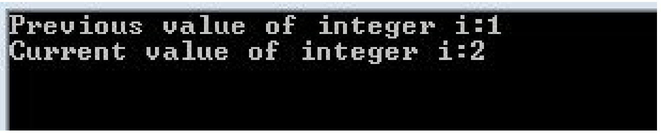
```
public static string GetNextName(ref int id){ string returnText = "Next-" + id.ToString();   id += 1; return returnText; }



  static void Main(string[] args)
  {

  int i = 1;
  Console.WriteLine("Previous value of integer i:" + i.ToString());
  string test = GetNextName(ref i);
  Console.WriteLine("Current value of integer i:" + i.ToString());
}
```

Output



```
Previous value of integer i:1
Current value of integer i:2
```

Out Keyword

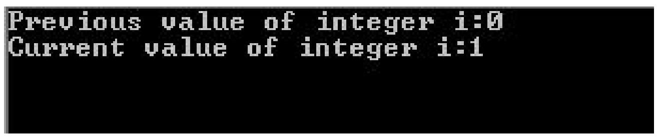
The out keyword passes arguments by reference. This is very similar to the ref keyword.

Example Code

```
public static string GetNextNameByOut(out int id)
{
    id = 1;
    string returnText = "Next-" + id.ToString();
    return returnText;
}

static void Main(string[] args)
{
    int i = 0;
    Console.WriteLine("Previous value of integer i:" + i.ToString());
    string test = GetNextNameByOut(out i);
    Console.WriteLine("Current value of integer i:" + i.ToString());
}
```

Output



```
Previous value of integer i:0
Current value of integer i:1
```

Ref Vs Out

Ref	Out
The parameter or argument must be initialized first before it is passed to ref.	It is not compulsory to initialize a parameter or argument before it is passed to an out.
It is not required to assign or initialize the value of a parameter (which is passed by ref) before returning to the calling method.	A called method is required to assign or initialize a value of a parameter (which is passed to an out) before returning to the calling method.
Passing a parameter value by Ref is useful when the called method is also needed to modify the pass parameter.	Declaring a parameter to an out method is useful when multiple values need to be returned from a function or method.
It is not compulsory to initialize a parameter value before using it in a calling method.	A parameter value must be initialized within the calling method before its use.
When we use REF, data can be passed bi-directionally.	When we use OUT data is passed only in a unidirectional way (from the called method to the caller method).
Both ref and out are treated differently at run time and they are treated the same at compile time.	
Properties are not variables, therefore it cannot be passed as an out or ref parameter.	

Ref / Out keyword and method Overloading

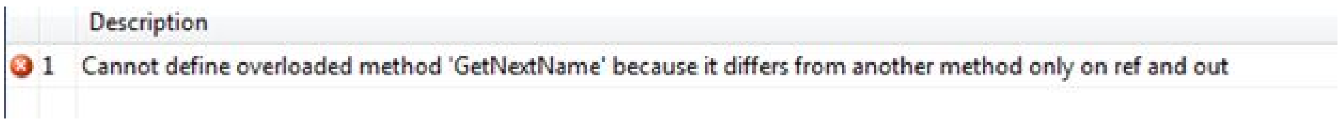
Both ref and out are treated differently at run time and they are treated the same at compile time, so methods cannot be overloaded if one method takes an argument as ref and the other takes an argument as an out.

Example code

```
public static string GetNextName(ref int id)
{
    string returnText = "Next-" + id.ToString();
    id += 1;
    return returnText;
}

public static string GetNextName(out int id)
{
    id = 1;
    string returnText = "Next-" + id.ToString();
    return returnText;
}
```

Output when the code is compiled:



```
    Description
1   Cannot define overloaded method 'GetNextName' because it differs from another method only on ref and out
```

However, method overloading can be possible when one method takes a ref or out argument and the other takes the same argument without ref or out.

Example Code

```
public static string GetNextName(int id)
{
    string returnText = "Next-" + id.ToString();
    id += 1;
    return returnText;
}
public static string GetNextName(ref int id)
{
    string returnText = "Next-" + id.ToString();
    id += 1;
    return returnText;
}
```

Summary

The out and ref keywords are useful when we want to return a value in the same variables as are passed as an argument.