

C# Language Proposal for 'out' Parameters

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-04/2625.html>

From: C# Learner (*csharp_at_learner.here*)

Date: 04/10/04

Date: Sat, 10 Apr 2004 08:04:15 +0100

Note

Please use a fixed-width font to view this, such as Courier New.

Problem

When passing a parameter with the modifier 'out', it is necessary to write two statements:
- one for the declaration of the variable due to receive the value; and
- one for the method call, where the variable is passed as a parameter.

Example:

```
static void Main()
{
    int i;
    if (TryGetValue(out i)) {
        Console.WriteLine("Value is: {0}.", i);
    }
}
```

This is inconsistent with the traditional way of returning values from a method.

Example:

```
static void Main()
{
    int i = GetValue();
    Console.WriteLine("Value is: {0}.", i);
}
```

Solution

The solution to this problem would be to allow the declaration of the variable to appear within the statement where it's being passed as an 'out' parameter.

Example:

```
static void Main()
{
    if (TryGetValue(out int i)) {
        Console.WriteLine("Value is: {0}.", i);
    }
}
```

Benefits

- (a) Method bodies can benefit from one less statement for each time this technique is used, promoting readability.
- (b) C# becomes more consistent since returning values using both the traditional way, and using 'out' parameters, is uniform. Again, this promotes readability.

microsoft.public.dotnet.languages.csharp: C# Language Proposal for 'out' Parameters

Example:

```
// traditional way
static void Main()
{
    int i = GetANumber();
    DoSomethingWith(i);
}
// using an 'out' parameter, with the aforementioned syntax
static void Main()
{
    if (TryGetANumber(out int i)) {
        DoSomethingWith(i);
    }
}
```

End

Comments?

Regards,

C. S. Learner