索引属性改进了在 C# 编程中使用具有参数的 COM 属性的方式。将索引属性与 Visual C# 2010 中引入的其他功能（如[命名参数和可选参数](https://msdn.microsoft.com/zh-cn/library/dd264739.aspx)、新的 [dynamic](https://msdn.microsoft.com/zh-cn/library/dd264741.aspx) 类型以及[嵌入的类型信息](https://msdn.microsoft.com/zh-cn/library/dd409610.aspx)）一起使用，可增强 Microsoft Office 编程功能。

在早期版本的 C# 中，只有在 **get** 方法不具有任何参数且 **set** 方法有且仅有一个值参数时，才能将方法作为属性进行访问。但是，并非所有 COM 属性都符合上述限制。例如，Excel 的 [Range](http://go.microsoft.com/fwlink/?LinkId=166053) 属性具有一个 **get** 访问器，该访问器需要一个表示范围名称的参数。以前，由于您无法直接访问 **Range** 属性，因此您必须改用 **get\_Range** 方法，如以下示例所示。

// Visual C# 2008 and earlier.

var excelApp = new Excel.Application();

// . . .

Excel.Range targetRange = excelApp.get\_Range("A1", Type.Missing);

利用索引属性，您可以改为编写以下代码：

// Visual C# 2010.

var excelApp = new Excel.Application();

// . . .

Excel.Range targetRange = excelApp.Range["A1"];

| **https://wizardforcel.gitbooks.io/msdn-csharp/content/img/note.jpg 注意** |
| --- |
| 上面的示例还使用[可选参数](https://msdn.microsoft.com/zh-cn/library/dd264739.aspx)功能（在 Visual C# 2010 中引入），此功能可使您忽略 **Type.Missing**。 |

与此类似，若要在 Visual C# 2008 及更早版本中设置 [Range](http://go.microsoft.com/fwlink/?LinkId=179211) 对象的 **Value** 属性的值，需要两个参数。一个参数为用于指定范围值的类型的可选形参提供实参。另一个参数提供 **Value** 属性的值。在 Visual C# 2010 之前的版本中，C# 仅允许一个参数。因此，您必须使用 **set\_Value** 方法，或者使用另一个属性 [Value2](http://go.microsoft.com/fwlink/?LinkId=166050)，而不能使用常规的 set 方法。下面的示例演示了这些技术。这两种技术都将 A1 单元格的值设置为 Name。

// Visual C# 2008.

targetRange.set\_Value(Type.Missing, "Name");

// Or

targetRange.Value2 = "Name";

利用索引属性，您可以改为编写以下代码。

// Visual C# 2010.

targetRange.Value = "Name";

您不能创建自己的索引属性。该功能只支持使用现有的索引属性。

下面的代码显示一个完整的示例。有关如何设置访问 Office API 的项目的更多信息，请参见[如何：通过使用 Visual C# 功能访问 Office 互操作对象（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/dd264733.aspx)。

// You must add a reference to Microsoft.Office.Interop.Excel to run

// this example.

using System;

using Excel = Microsoft.Office.Interop.Excel;

namespace IndexedProperties

{

class Program

{

static void Main(string[] args)

{

CSharp2010();

//CSharp2008();

}

static void CSharp2010()

{

var excelApp = new Excel.Application();

excelApp.Workbooks.Add();

excelApp.Visible = true;

Excel.Range targetRange = excelApp.Range["A1"];

targetRange.Value = "Name";

}

static void CSharp2008()

{

var excelApp = new Excel.Application();

excelApp.Workbooks.Add(Type.Missing);

excelApp.Visible = true;

Excel.Range targetRange = excelApp.get\_Range("A1", Type.Missing);

targetRange.set\_Value(Type.Missing, "Name");

// Or

//targetRange.Value2 = "Name";

}

}

}

**请参阅**

[命名实参和可选实参（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/dd264739.aspx)

[dynamic（C# 参考）](https://msdn.microsoft.com/zh-cn/library/dd264741.aspx)

[使用类型 dynamic（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/dd264736.aspx)

[如何：在 Office 编程中使用命名参数和可选参数（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/dd264738.aspx)

[如何：通过使用 Visual C# 功能访问 Office 互操作对象（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/dd264733.aspx)

[演练：Office 编程（C# 和 Visual Basic）](https://msdn.microsoft.com/zh-cn/library/ee342218.aspx)