# Workbook.RegisterCustomFunction

**Introduced in build 9.1**

## Description

The RegisterCustomFunction method registers an IFunction object with the workbook to be used when calculating a specific custom Excel formula.

---

**C#**

```csharp
public void RegisterCustomFunction(string customFunctionName, IFunction
customFunction)
```

---

**vb.net**

```vbnet
Public Sub RegisterCustomFunction(customFunctionName As String, customFunction As
IFunction)
```

---

⚠ **Disabling Excel Calculation**
Registering a formula allows ExcelWriter to calculate a function a certain way. However, once the workbook opens inside of Microsoft Excel, Excel will calculate the formula again with its own calculation engine. This means that if you registered a function name that does not exist in Excel, your formula will evaluate to #NAME? once Excel recalculates it. If you overwrote a formula with you own implementation, the formula will be recalculated using the implementation that is built into Excel. To keep Excel from re-calculating the formulas upon opening the workbook or upon changing values in the workbook, go to File -> Options -> Formulas. Under the "Calculation Options" header, there is an option called "Workbook Calculation". Please select the "Manual" option and uncheck the "Recalculate workbook before saving" checkbox. Now Excel will not re-calculate formulas. These settings will allow you to preserve the values that ExcelWriter calculated.

## Parameters

### *customFunctionName*

The name of the Excel Function you want to bind your custom IFunction logic to.

### *customFunction*

The custom logic you want to use for a specific Excel formula. You implement this logic by using the IFunction interface

## Examples

**C#**

```csharp
//Create an ExcelApplication object
ExcelApplication xla = new ExcelApplication();

//Open a workbook containing formulas
Workbook wb = xla.Open("FormulasWorkbook.xlsx);

// Create an instance of the custom function implementation
IFunction myCustomFunction = new MyCustomFunction();
//Register the custom function
wb.RegisterCustomFunction("MyCustomFunction", myCustomFunction);
```

**VB**

```vb
'Create an ExcelApplication object
Dim xla As New ExcelApplication

'Open a workbook containing formulas
Dim wb As Workbook = xla.Open("formulasWorkbook.xlsx")

'Create an instance of the custom function implementation
Dim myCustomFunction As IFunction = new MyCustomFunction()

'Register the custom function
wb.RegisterCustomFunction("MyCustomFunction", myCustomFunction)
```

## Exceptions

### *SAException*

If the custom function has already been registered with the workbook.