

# Formulas

ExcelApplication supports all math, string, boolean, time, statistical, database, lookup and reference formulas or functions that are part of Excel. ExcelWriter does not calculate formulas at run-time; all formulas are calculated when the workbook is opened in Excel.

To insert a formula in a cell, use the property `Cell.Formula`. Begin the formula string with `=`, as you would when entering a formula in Microsoft Excel:

```
sheet1.Cells["A25"].Formula = "=SUM(A4:A24)";
```

ExcelWriter allows you to use values from a different sheet when assigning a formula to a cell, as in the following examples:

```
sheet1.Cells[5,3].Formula = "=SUM(Sheet1!A1:A3,Sheet3!A1:A5)";
```

The property `Cell.Name` returns a cell's Excel-style name (e.g., "B5"). This property can be useful if you want to build a formula string using row and column indexes, rather than cell names, for example:

```
String formula = "=SUM(" + sheet1.Cells[3, 1].Name +  
    ":" + sheet1.Cells[23, 1].Name + ")";  
sheet1.Cells[24, 1].Formula = formula;
```

## Example

Here is an example that demonstrates how to use Formulas with ExcelApplication:

```
//--- Create ExcelApplication, a Workbook, and a Worksheet  
ExcelApplication xlw = new ExcelApplication();  
Workbook wb = xlw.Create(ExcelApplication.FileFormat.Xlsx);  
Worksheet sheet1 = wb.Worksheets[0];  
  
//--- Write some values into cells for the  
//--- formulas to compute  
sheet1.Cells["A1"].Value = 1;  
sheet1.Cells["A2"].Value = 2;  
sheet1.Cells["A3"].Value = 3;  
  
//--- Write a formula into a cell  
//--- Formula will be calculated  
//--- when opened in Excel  
sheet1.Cells["A4"].Formula = "=SUM(A1:A3)";  
  
//--- Save the workbook  
xlw.Save(wb, "Generated.xlsx");
```

## Code Sample

This sample shows many of the core features of ExcelApplication, including the use of formulas.

[\[C#\]](#) | [\[VB.NET\]](#)