



# Programming with Smart Templates using sourceCREATOR™ 1.x

A few very basic examples of writing and using Smart Templates

CopyRight © 2018 - Ramin Zaghi

visualSilicon™

## TOC

<b>Basics of writing Smart Templates and using sourceCREATOR™ command line tools</b>	<b>2</b>
Example 1 - very basic use	2
Example 2 - using a placeholder	2
Example 3 - put your value in a different component file	3
Example 4 - using attributes and the "output" attribute	4
Example 5 - Generating a simple calculator program in C#	5

## Basics of writing Smart Templates and using sourceCREATOR™ command line tools

### Example 1 - very basic use

```
== my_component.txt ==  
Hello  
My name is sourceCREATOR.  
The end.  
== == == ==  
$ scc my_component.txt  
$ scgen my_component
```

The output on the console will be:

```
Hello  
My name is sourceCREATOR.  
The end.
```

### Example 2 - using a placeholder

```
== my_component.txt ==  
Hello  
My name is /*< THE_NAME />*/.  
The end.  
/*< THE_NAME >*/  
sourceCREATOR  
/*</ THE_NAME >*/  
== == == ==  
$ scc my_component.txt  
$ scgen my_component
```

The output on the console will be:

```
Hello  
My name is sourceCREATOR.  
The end.
```

### Example 3 - put your value in a different component file

```
== my_component.txt ==  
Hello  
My name is /*< THE_NAME />*/.  
The end.  
/*< + another_component />*/  
== == == ==
```

```
== another_component.txt ==  
/*< THE_NAME >*/  
sourceCREATOR  
/*</ THE_NAME >*/  
== == == ==
```

```
$ scc another_component.txt my_component.txt  
$ scgen my_component
```

The output on the console will be:

```
Hello  
My name is sourceCREATOR.  
The end.
```

## Example 4 - using attributes and the "output" attribute

```

== my_component.txt ==
Hello
The text is `/*< THE_TEXT age="2" />*/"!
The end.
/*< + another_component />*/
/*< SOME_OTHER_TEXT output="path\output.txt" />*/
== == == ==

== another_component.txt ==
/*< THE_TEXT >*/
sourceCREATOR is /*< @ age />*/ years old
/*</ THE_TEXT >*/

/*< SOME_OTHER_TEXT >*/
This is in a new output file!
/*</ SOME_OTHER_TEXT >*/
== == == ==

$ scc my_component.txt another_component.txt
$ scgen my_component

```

The output on the console will be:

```

Hello
The text is "sourceCREATOR is 2 years old"!
The end.

```

And the output to the file "`path\output.txt`" will be:

```

This is in a new output file!

```

## Example 5 - Generating a simple calculator program in C#

Please refer to [this shared zip file](#):

[sourceCREATOR\\_The\\_Calculator\\_Example.zip](#)

```
$ scc operator_component.scc my_app.scc program.scc
$ OUTPUT_HELP_FILE="help\my_help.html" scgen my_app
```

The output on the console will be:

```
using System;

namespace CS_SAMPLE
{
    class MainClass
    {
        private static double AddNums(double a, double b)
        {
            return a + b;
        }

        private static double SubtractNums(double a, double b)
        {
            return a - b;
        }

        private static double DivideNums(double a, double b)
        {
            return a / b;
        }

        private static double MultiplyNums(double a, double b)
        {
            return a * b;
        }

        private static double AreNumsEquals(double a, double b)
        {
            return a = b;
        }

        public static void Main(string[] args)
        {
            if (args.Length <= 0) return;

            if (args[0] == "help")
            {
                Console.WriteLine("Usage:");
                Console.WriteLine("CS_SAMPLE a + b   Add two numbers");
                Console.WriteLine("CS_SAMPLE a - b   Subtract two numbers from each other");
                Console.WriteLine("CS_SAMPLE a / b   Divide two numbers");
                Console.WriteLine("CS_SAMPLE a * b   Multiply two numbers");
                Console.WriteLine("CS_SAMPLE a = b   Test equality");
                Console.WriteLine("CS_SAMPLE help    Show this help");
                Environment.Exit(0);
            }
        }
    }
}
```

```
if (args.Length < 3) return;
double a = Double.Parse(args[0]);
double b = Double.Parse(args[2]);
string op = args[1];

switch (op)
{
    case "+":
        Console.WriteLine(a + " " + op + " " + b + " = " + AddNums(a,b));
        break;
    case "-":
        Console.WriteLine(a + " " + op + " " + b + " = " + SubtractNums(a,b));
        break;
    case "/":
        Console.WriteLine(a + " " + op + " " + b + " = " + DivideNums(a,b));
        break;
    case "*":
        Console.WriteLine(a + " " + op + " " + b + " = " + MultiplyNums(a,b));
        break;
    case "=":
        Console.WriteLine(a + " " + op + " " + b + " = " + AreNumsEquals(a,b));
        break;

    default:
        Console.WriteLine("Unknown command");
        break;
}
}
```

And the output to the file "`help\my help.html`" will be:

```
<b>// THIS IS A FEATURE: Add two numbers </b><br/>
<b>// THIS IS A FEATURE: Subtract two numbers from each other </b><br/>
<b>// THIS IS A FEATURE: Divide two numbers </b><br/>
<b>// THIS IS A FEATURE: Multiply two numbers </b><br/>
<b>// THIS IS A FEATURE: Test equality </b><br/>
```