

```

1 // The top of every source code file must include this line
2 #include "sierrachart.h"
3
4 // For reference, refer to this page:
5 //
6 https://www.sierrachart.com/index.php?page=doc/AdvancedCustomStudyInterfaceAndLanguage.ph
7 p
8
9 // This line is required. Change the text within the quote
10 // marks to what you want to name your group of custom studies.
11 SCDLLName("BB Subgraph Add")
12
13 //This is the basic framework of a study function. Change the name 'TemplateFunction'
14 // to what you require.
15 SCSFExport scsf_SubgraphAddition(SCStudyInterfaceRef sc)
16 {
17     SCSubgraphRef Subgraph_Result = sc.Subgraph[0];
18
19     SCInputRef Input_Data = sc.Input[0];
20     SCInputRef Input_AmountToAdd = sc.Input[1];
21     SCInputRef Input_DrawZeros = sc.Input[2];
22     SCInputRef Input_AddToZeroValuesInBasedOnStudy = sc.Input[3];
23     SCInputRef Input_NumberOfForwardProjectionBars = sc.Input[6];
24
25     if (sc.SetDefaults)
26     {
27         sc.GraphName = "BB Study Subgraph Add";
28
29         sc.StudyDescription = "This study adds the specified value to the selected
30         study Subgraph. To select the study to use, set the 'Based On' study setting to
31         that study and set the 'Input Data' input to the specific study Subgraph in
32         that study to Add the value to.";
33
34         sc.AutoLoop = 0; //Manual looping
35
36         Subgraph_Result.Name = "Result";
37         Subgraph_Result.DrawStyle = DRAWSTYLE_LINE;
38         Subgraph_Result.PrimaryColor = RGB(0,255,0);
39
40         Input_Data.Name = "Input Data";
41         Input_Data.SetInputDataIndex(0);
42
43         Input_AmountToAdd.Name = "Amount to Add";
44         Input_AmountToAdd.SetFloat(0);
45
46         Input_DrawZeros.Name = "Draw Zeros";
47         Input_DrawZeros.SetYesNo(false);
48
49         Input_AddToZeroValuesInBasedOnStudy.Name = "Add to Zero Values in Based On
50         Study";
51         Input_AddToZeroValuesInBasedOnStudy.SetYesNo(true);
52
53         Input_ProjectForward.Name = "Forward Project Current Price Line";
54         Input_ProjectForward.SetYesNo(false);
55
56         return;
57     }
58
59     Subgraph_Result.DrawZeros = Input_DrawZeros.GetYesNo();
60
61     int CalculationStartIndex = sc.GetCalculationStartIndexForStudy();
62
63     for (int Index = CalculationStartIndex; Index < sc.ArraySize; Index++)
64     {
65         float BasedOnStudySubgraphValue =
66         sc.BaseData[Input_Data.GetInputDataIndex()][Index];
67
68         if (Input_AddToZeroValuesInBasedOnStudy.GetYesNo() == 0 &&

```

```
62     BasedOnStudySubgraphValue == 0.0)
63     {
64         Subgraph_Result[Index] = 0.0;
65     }
66     else
67     {
68         Subgraph_Result[Index] = BasedOnStudySubgraphValue +
69             Input_AmountToAdd.GetFloat ();
70     }
71     if (Input_ProjectForward.GetYesNo ())
72     {
73         int NumberOfBars = Input_NumberOfForwardProjectionBars.GetInt ();
74         Subgraph_CurrentPriceLine.ExtendedArrayElementsToGraph = NumberOfBars;
75         for(int Index = 0; Index < NumberOfBars; Index++)
76         {
77             Subgraph_CurrentPriceLine[sc.ArraySize + Index] = LineValue;
78         }
79         r_PriorLineStartIndex = sc.ArraySize;
80     }
81     else
82     Subgraph_CurrentPriceLine.ExtendedArrayElementsToGraph = 0;
83 }
84 }
85 sc.EarliestUpdateSubgraphDataArrayIndex = CalculationStartIndex;
86 }
87 }
88 }
89 }
90 }
91 }
```