

# Using ACSIL Study Calculation Functions

- [Introduction](#)
  - [Example Source Code](#)
- 

## Introduction

---

This page discusses how you can use the ACSIL (Advanced Custom Study Interface and Language) study calculation functions that require input arrays such as [sc.SimpleMovAvg](#), `sc.MACD`, ..., and pass them the result of an intermediate calculation.

When you simply pass an existing array such as `sc.BaseData[SC_LAST]` as the first parameter to these functions, then this is very easy. However, what if you want to perform some intermediate calculation, before passing it to one of these study calculation functions?

If you are familiar the Sierra Chart **Spreadsheet Studies** which use a Spreadsheet for the calculations, then we will use this as an example to demonstrate what we are explaining here.

For example, say that we want to add the value of 10 to a bar Last trade price and then calculate a 10 bar moving average of those values. This will need to be done with two separate formulas and columns.

First we add 10 and then we calculate the average. Spreadsheet column E contains the bar Last prices. In Column Z (you can use any available formula column that does not have a visible Draw Style), we will use this formula to add 10: `=E3+10`. In Column K we will use this formula to calculate the average:

`=AVERAGE(Z3:Z12)`.

Since the **AVERAGE** function requires an array, this is specified through a reference to a range of cells, we first had to create a column with the data we want to calculate the average of and we did this in column Z. Therefore, you can see that you have to first perform the intermediate calculation in column Z or any available column, and then calculate the average in column K.

When using the ACSIL, we also need to do, in this example, a two-step process. The way that we will accomplish this, is by using an available [internal extra array](#) in a Subgraph or a [Subgraph Data array](#) and filling it in with the a result of our intermediate calculation before passing it to one of the study calculation functions.

Below is the complete code example for this. In this example we first take the Last trade price and add 10 to it and assign the result into one of the available internal extra arrays of a Subgraph.

Since our result is going to be placed into `Subgraph[0]`, we are using one of those internal extra arrays. This keeps our code very organized by using the same Study Subgraph object.

We start by using internal array at index 11 (`sc.Subgraph[0].Arrays[11]`). Since indexing is zero based, 11 refers to array 12. There are `MAX_STUDY_EXTRA_ARRAYS` extra arrays per Subgraph.

We are using the last one because some study calculation functions will also use the internal extra arrays for their calculations so we do not want our result overwritten.

The documentation for each of the available study calculation functions explains how many arrays are used by a study calculation function. So you know which ones you can safely use.

## Example Source Code

```
SCSFExport scsf_IntermediateStudyCalculationsUsingArrays(SCStudyInterfaceRef sc)
{
    SCSubgraphRef Average = sc.Subgraph[0];
    SCInputRef Input_Length = sc.Input[0];

    if (sc.SetDefaults)
    {
        sc.GraphName = "Intermediate Study Calculations Using Arrays";

        sc.StudyDescription = "For more information about this see the Using ACSIL Study Calculation Functions document";

        sc.AutoLoop = 1;

        Average.Name = "Average";

        Average.DrawStyle = DRAWSTYLE_LINE;

        Input_Length.Name = "Moving Average Length";

        Input_Length.SetInt(10);
        Input_Length.SetIntLimits(1, 9999999);

        return;
    }

    // Here we will add 10 to the sc.BaseData Last value at the current index
    // and compute an average of it. Since the moving average function we
    // will be calling requires an input array and not a single value, we
    // will use one of the internal extra arrays on a Subgraph to hold
    // this intermediate calculation. This internal extra array could be
    // thought of as a Spreadsheet column where you are performing intermediate
    // calculations. We will use one of the internal extra arrays that is
    // part of the Subgraph we are using to hold the output from the moving
    // average study calculation function we will be calling next. Although
    // any Subgraph internal extra array or even a Subgraph Data array
    // could be used.

    SCFloatArrayRef Last = sc.BaseData[SC_LAST];

    SCFloatArrayRef SubgraphExtraArray = Average.Arrays[0];

    SubgraphExtraArray[sc.Index] = Last[sc.Index] + 10;

    // In this function call we are passing this internal extra array and
    // we also pass in, Average (sc.Subgraph[0]), to receive the result at the
    // current index (sc.Index).

    sc.SimpleMovAvg(SubgraphExtraArray, Average, Input_Length.GetInt());
}
```

