IEC61850 Array Type Elements

Using Arrays in PIS10

Application Note

Introduction

Arrays are a data type in IEC61850 configuration files; this document will describe how to use these arrays in PIS10 function calls.

Where to use Arrays

Arrays can be used on any of the following function calls:
IEC61850_Update(…)
IEC61850_Write(…)
IEC61850_Read(…)

Arrays can also be used in the following callbacks triggered by the stack:
IEC61850_UpdateCallback
IEC61850_ReadCallback
IEC61850_WriteCallback

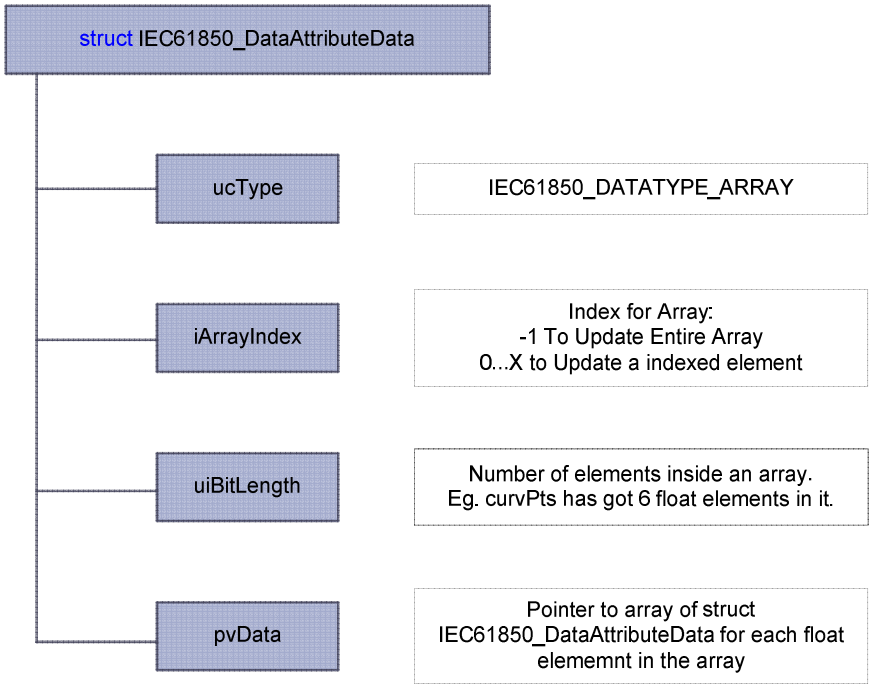
There are 2 ways to use Arrays in function calls, the Data Attribute Data structure that is passed into, or out from, the stack will have an index value. This is only used for arrays, if the value in this field is a positive, or zero, number less than the size of the array, it refers to the index that is being passed. If the value is -1 it means that the entire array is to be passed.

The PIS10 stack only supports Arrays in MMS communications. This includes reporting, but does not support Arrays being transmitted in Last App Error, Command Termination, GOOSE, or Sampled Values communications.

Creating the structure to pass into API calls

The PIS10 API accepts arrays as a variation of the Data Attribute Data structure used in API function calls.
The key difference is that the bitLength passed into the Array structure is repurposed to specify the number of elements being pointed to by the pvData.

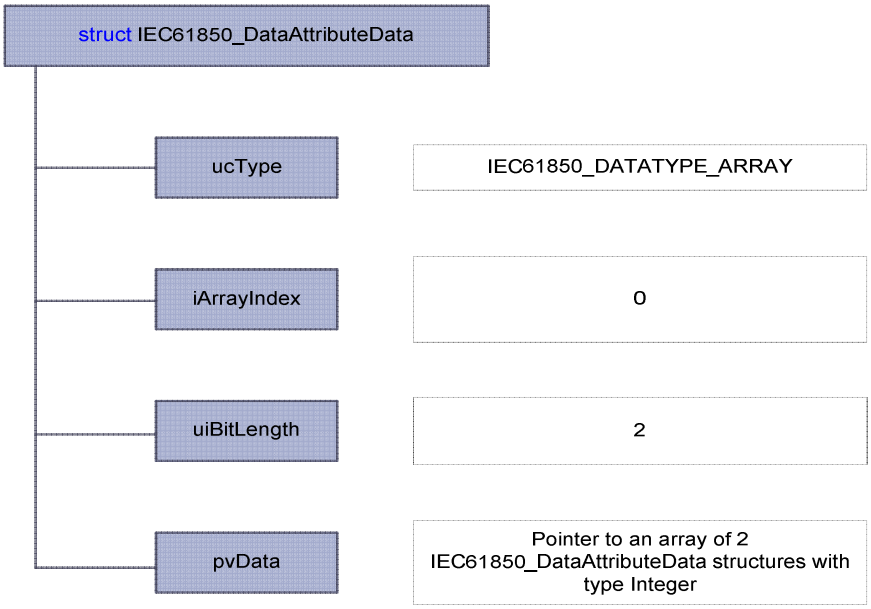
The image below shows the basic form of the Data Attribute Data structure at the top level.
The ucType is IEC61850_DATATYPE_ARRAY, this specifies that the structure that follows is to be used as an array.
The iArrayIndex is the individual element you are updating, or -1 for the whole array.
The uiBitLength element is the number of elements pointed to by pvData.
pvData is a pointer to a series of IEC61850_DataAttributeData structures containing the values for each point in the array.



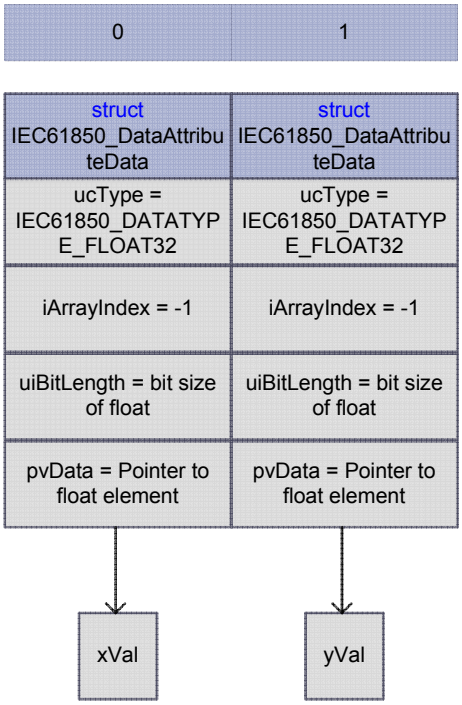
The list that the pvData points to is a series of IEC61850_DataAttributeData structures that contain pointers to the individual values to be stored in the array.

In the case of a curvPts type array, each element of the array has an x-val and a y-val. This means that to update any one point in the array, you will need to pass in 2 elements: One for x-val and one for y-val.

If you were updating element 0 (first element of the array) you would pass in the following top level structure:

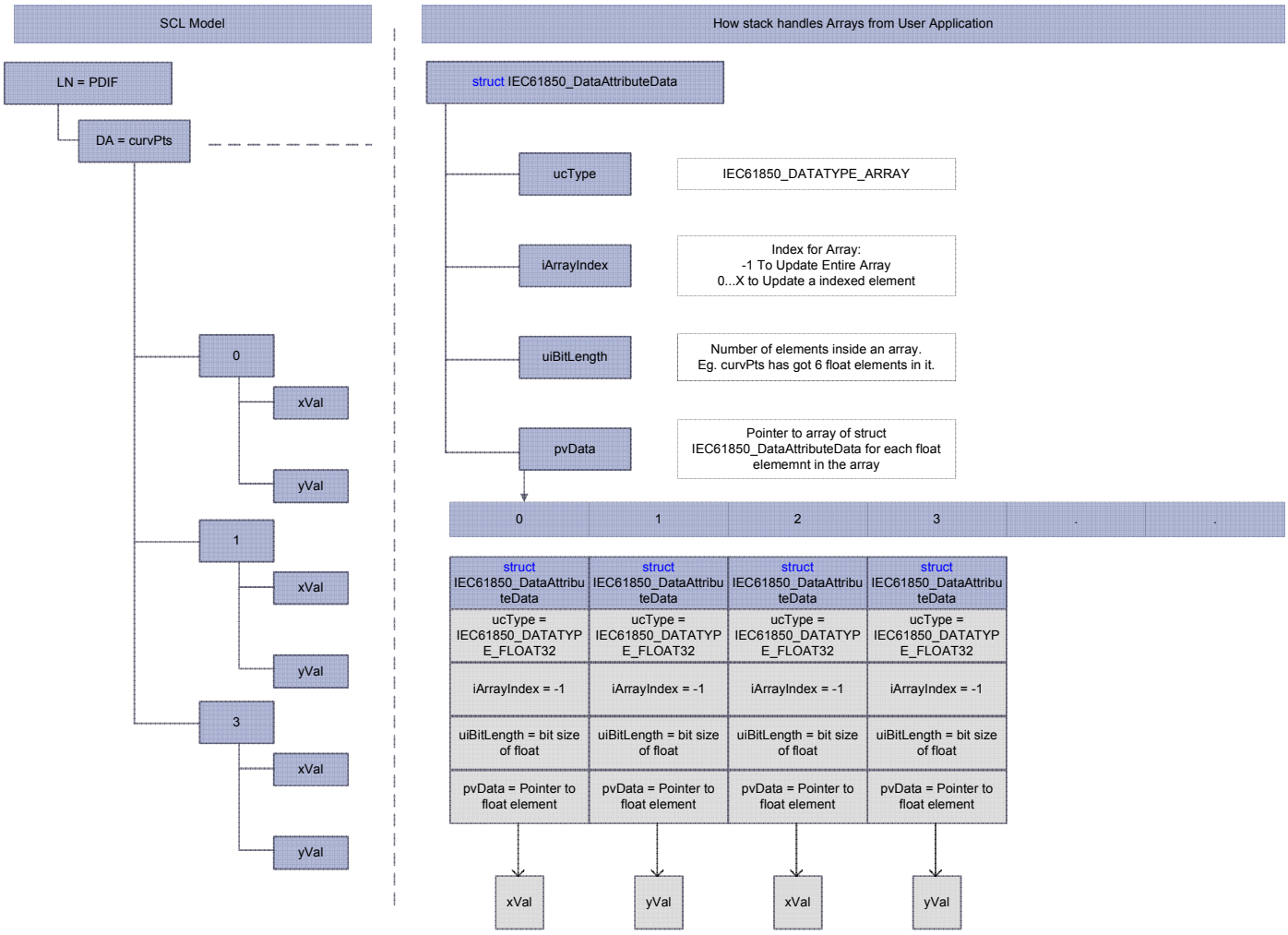


The pvData would point to an array containing all the elements needed to update the index specified in the top level Data Attribute Data structure. Continuing the example above, the pvData will point to an array of data points looking like this:



If the index was set as -1, the pvData would need to point to an array containing a number of elements equal to (the number of elements in the array multiplied by the number of elements in each element of each array)

The below image is the complete model of the data and its associated data structure in PIS10:



If you need assistance

All technical questions must be sent to our support email address: **support@systemcorp.com.au**
Upon receiving your question(s), it will get logged in our support system and you will receive and acknowledgement which will include a tracking number(s), e.g. M#040321. Please refer to your tracking number(s) when you are following up about enquiry.