Technical Note 14: DAC Variable Offsets

This note describes how the GPT views the static data managed by the user application (DAC). The GPT views static data for a DAC object as an array. The array is accessed by offset or index. Arrays are always zero based; so the first variable for a DAC object is always found at offset 0. For example, the following table shows how the GPT views five analog input variables.

Offset	DAC Variable
0	Analog 1
1	Analog 2
2	Analog 3
3	Analog 4
4	Analog 5

The GPT uses the DAC property IPROP_DACOBJ_STATIC_COUNT to obtain the dimension of the static array allocated for an object. During configuration of the system, the user application assigns configuration information to each variable. For most protocols, a property of a variable is its ID or identification. So, for example, the user could configure the above array as follows:

Offset	DAC Variable
0	Analog 1, ID = 1
1	Analog 2, ID = 100
2	Analog 3, ID = 1001
3	Analog 4, ID = 1002
4	Analog 5, ID = 64000

In the above example the user has configured the five analog input variables to have the IDs 1, 100, 1001, 1002, and 64000 respectively.

Note: *For all protocols, the GPT assumes the user application maintains a sorted list of variables. That is, ai[n].Id > ai[m].Id if n > m.*

Whenever possible, the GPT avoids accessing DAC variables by ID and uses the variable's offset or array index. If the GPT needs to access variables by ID, the user application (DAC) provides a property and an API to allow the GPT to search for a specific ID. The DAC property IPROP_DACPNT_ID returns the IDs for the requested DAC variables. In the above example, if the GPT calls the DacPntGet service for the analog input object with property IPROP_DACPNT_ID and offset 2, the user application returns the value 1001. Each DAC object may also support the DAC Find API. The Find API

searches an array for a specific ID and returns the offset in the array where the ID was found. The Find API returns the following result codes based on the ID search:

- **0.** The id was found in the static array.
- **Greater than 0**. The ID was not found in the static array, but a variable exists with a larger ID.
- **Less than 0.** The id was not found in the static array, and no variable exists with an ID that is larger.

Using the array above as an example, the Find operates as follows:

Find(id = 1) return code = 1 @ offset 0
Find(id = 70000) return code = -1
Find(id = 50000) return code = 0 @ offset 4

With the IEC protocol, the IECPT uses the DAC Find API on output variables to translate the variable identification it receives from the remote station into an offset into the object static array. The IECPT does not use the find service for input variables. With input variables, the IECPT reads the variables by offset and uses the ID field in the ANALOGINPUT.Id structure to decide whether the response ASDU can be transmitted sequentially.