



Applied Systems Engineering, Inc.

**Technical Note #29**  
**DAC Static/Event Writes in Master Mode**

## About DAC Static/Event APIs

Consider the following example IEC example. The user application supports two double variables as follows:

DAC offset	Double Id (IEC Object Information Address)
0	1001
1	5000

The two variables are with Object Information Address 1001, and 5000 are addressed using variable offsets 0 and 1 respectively.

### DAC Events Writes

Event writes are position independent. When the GPT needs to write a double event it calls the DAC Write Events API. The API is passed a count of events to write and a buffer containing the event data. The DAC offset is not used in the event write so the GPT never calls the DAC Find API when performing event writes.

### DAC Static Writes

Static writes are position dependent. The GPT writes to a specific range of DAC offsets when writing static data. In order to determine the position of the write the GPT will call the DAC Find service to translate the variable ID e.g. 1001 to an offset in the DAC variable array. When the GPT writes the variable with id 1001 it first calls the DAC Find API for the object. The find service translates 1001 to offset 0 the location of the variable within the static array of double variables. The GPT then calls DacWriteBinaryInput with start = 0, end = 0, and a buffer containing the data to be written. To write to object information address 5000 the procedure is the same. Find is called to translate 5000 to offset 1. DacWriteBinaryInput is called with start = 1, end = 1, and a buffer containing the data to write. The DAC Find API for an object is always called before the static write is performed.

### Position Independent Static Writes

Since the data written by the GPT contains the object information address of the variable the user application can choose to treat static writes as if they are position independent. This is done as follows. The user constructs a DAC Find API that always returns TRUE (variable found). When the GPT calls the DAC Static Write Service the offset is ignored and the Id (information address) contained in the buffer is used to locate the variable. As an example consider the double points in the example above. Assume the user provided two services for doubles. DacFindDouble and DacWriteDouble. This routine would be defined as follows:

```
DacFindDouble( DACID id, DACOBJREF *offset)
{
    /* Ignore the id, always treat the find as successful */

    GPTUNUSED( id );
```

## Technical Note 29: DAC Static/Event Writes in Master Mode

```
    *offset = 0;
    return TRUE;
}

DacWriteDouble( DACOBJREF start, DACOBJREF end, LPBINARYINPUT buf )
{
    /* Start always 0, because in DacFindDouble *offset = 0 */
    /* Total number of points to write is end – start + 1 */
    /* or end + 1 since start always 0 */

    For( I=0;I<end+1;I++)
    {
        /* use the id of the variable 1001 or 5000 to decide where to store */
        /* store the variable */
        StoreDoubleVariable( buf[I].Id )
    }

    return I;
}
```