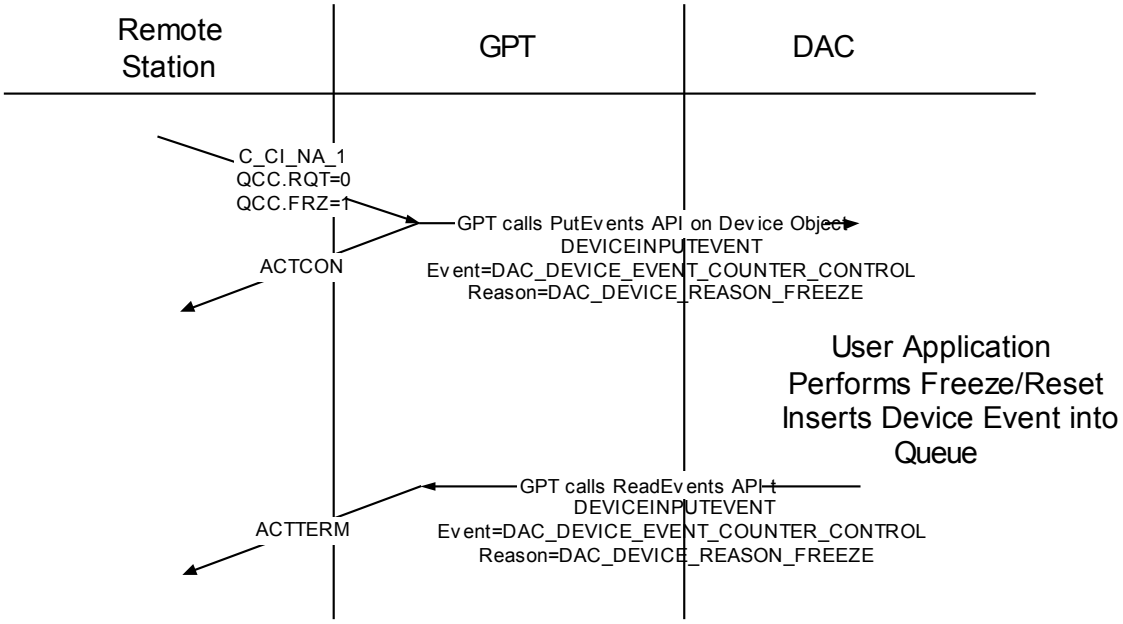
Technical Note #32
IEC Counter Interrogation

Counter Interrogation

This note discusses the interaction between the GPT and the user application when the remote station performs a counter interrogation. This note considers two cases. In case 1 counters are frozen or reset but not polled. In case 2 counters are frozen/reset and polled.

Case 1: Counter Freeze/Reset

This section examines a counter interrogation request in which counters are frozen/reset, but not polled. The next section will examine the case were counter data is polled. The interaction between the remote station, GPT and the user application is illustrated in the following figure:

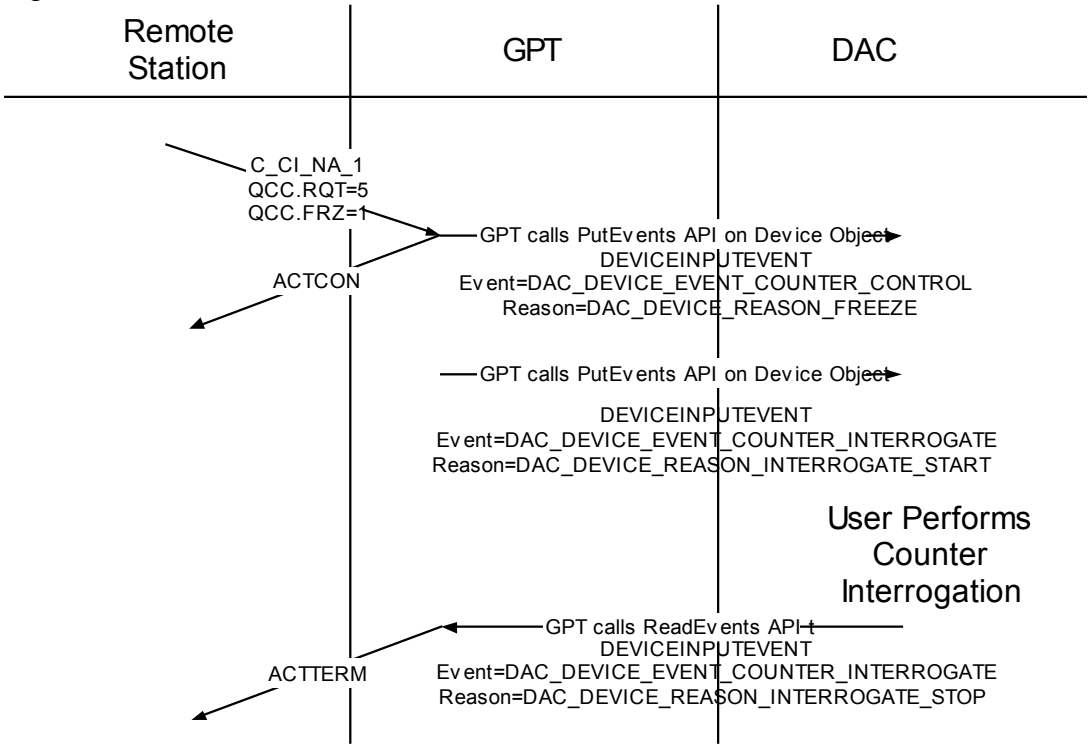


The counter freeze is processed as follows:

- The C_CI_NA_1 ASDU is received from a remote station. The GPT parses the ASDU and calls the DAC Put Event API on the device object. The event written is a DEVICEINPUTEVENT with event = DAC_DEVICE_EVENT_COUNTER_CONTROL and reason = DAC_DEVICE_REASON_FREEZE.
- The user application performs the freeze. When the freeze completes the user inserts into the event queue a DEVICEINPUTEVENT with event = DAC_DEVICE_EVENT_COUNTER_CONTROL and reason = DAC_DEVICE_REASON_FREEZE. When this event is read by the GPT a C_CI_NA_1 ASDU is generated with COT=ACTTERM.

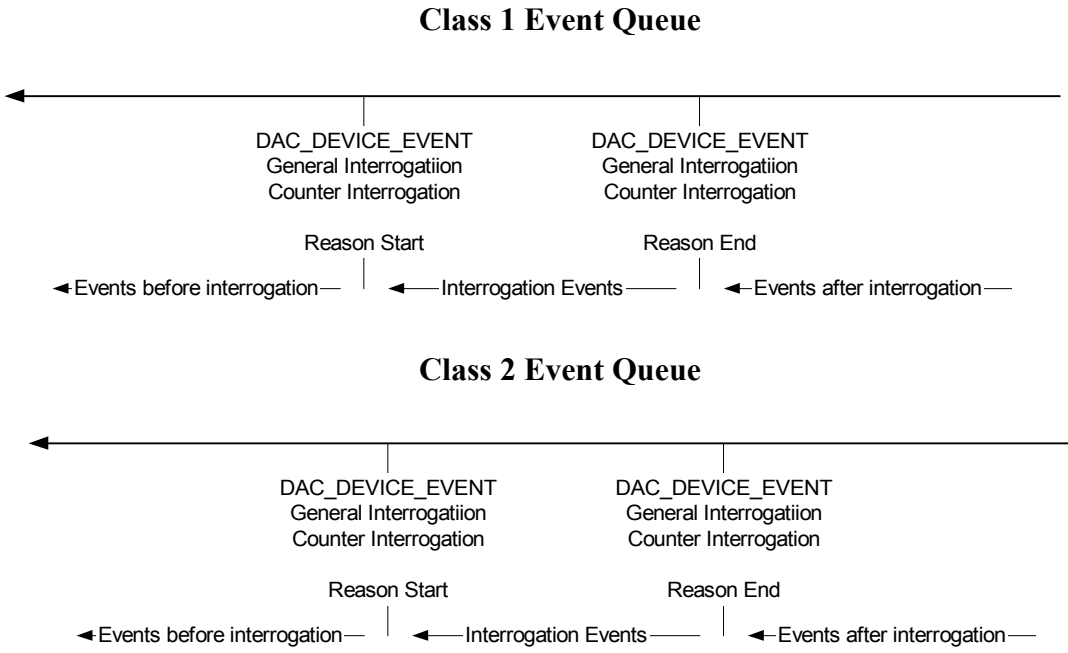
Case 2. Counter Interrogation with counters returned.

This section examines a counter interrogation request in which counters are frozen/reset, and polled.



The counter interrogation is processed as follows:

- The C_CI_NA_1 ASDU is received from a remote station. The GPT parses the ASDU and calls the DAC Put Event API on the device object. The event written is a DEVICEINPUTEVENT with event = DAC_DEVICE_EVENT_COUNTER_CONTROL and reason = DAC_DEVICE_REASON_FREEZE. The user performs all of the steps described above in case 1.
- The GPT calls the DAC Put Event API on the device object. The event written is a DEVICEINPUTEVENT with event = DAC_DEVICE_EVENT_COUNTER_INTERROGATE and reason = DAC_DEVICE_REASON_INTERROGATION_START. This is a signal to the user to start the interrogation. The user inserts markers into the class1 and class2 event queues to identify the interrogation events. The queues should be setup as follows:



- The markers (DEVICEINPUTEVENT) must be present in the queues even if there are no counter interrogation events for the data class. If counters are returned as class 1 data then there will be no interrogation events in the Class 2 queue, but the start and end interrogation events will still be present. Counter interrogation events consist of 0 or more COUNTERINPUTEVENT records.
- The Counter interrogation is terminated when the DEVICEINPUTEVENT with reason = DAC_DEVICE_REASON_INTERROGATION_STOP is read from both the class 1 and class 2 data queues. The GPT signals the end of interrogation by calling the DAC Put Event API on the Device object. The DEVICEINPUTEVENT with event = DAC_DEVICE_EVENT_COUNTER_INTERROGATE and reason = DAC_DEVICE_REASON_INTERROGATION_STOP. A C_CI_NA_1 ASDU is generated with COT=ACTTERM and transmitted to the remote station.

Example Code

This section will illustrate the above concepts using small code examples:

```
DacDeviceWriteEvent(LPDACENV ep,,, LPDEVICEINPUTEVENT p )
{
       switch(p->Event)
       {
       case DAC_DEVICE_EVENT_COUNTER_CONTROL :

              switch(  p->Reason )
              {
              case DAC_DEVICE_REASON_FREEZE :
              case DAC_DEVICE_REASON_FREEZE_RESET:
              case DAC_DEVICE_REASON_RESET :

                     /* User application performs freeze reset */
                     /* User inserts event into event queue to signal freeze
              complete */
```

```
                DacWriteQueue( pEnv, devId, QUEUE_PRIORITY_HIGH,
                       (DacVariable) device, off, p,
                       sizeof(DEVICEINPUTEVENT) );

                break;

        case DAC_DEVICE_EVENT_COUNTER_INTERROGATE:

               switch( p->Reason )
               {
               case DAC_DEVICE_REASON_INTERROGATE_START :


                       /* Class 1 Counter Interrogation */
                       ep->Class = 1;

                       p->Reason = DAC_DEVICE_REASON_INTERROGATE_START;

                       /* Write start marker into queue */
                       DacWriteQueue( pEnv, devId, QUEUE_PRIORITY_LOW,
                              (DacVariable) device, off, p,
                              sizeof(DEVICEINPUTEVENT) );

                       /* User inserts into queue all class 1 COUNTER events */
                       for(I=0;I<CounterCount;I++)
                              if ( send counter as class 1 )
                              DacWriteQueue( pEnv, devId, QUEUE_PRIORITY_LOW,
                                     (DacVariable) counter, i,
                                     counterevent, sizeof(COUNTERINPUTEVENT) );

                       p->Reason = DAC_DEVICE_REASON_INTERROGATE_STOP;

                       /* Write stop marker into queue */
                       DacWriteQueue( pEnv, devId, QUEUE_PRIORITY_LOW,
                              (DacVariable) device, off, p,
                              sizeof(DEVICEINPUTEVENT) );

                       /* Class 2 Counter Interrogation */
                       ep->Class = 2;

                       p->Reason = DAC_DEVICE_REASON_INTERROGATE_START;

                       /* Write Start Marker into queue */
                       DacWriteQueue( pEnv, devId, QUEUE_PRIORITY_LOW,
                              (DacVariable) device, off, p,
                              sizeof(DEVICEINPUTEVENT) );

                       /* User inserts into queue all class 2 COUNTER events */
                       for(I=0;I<CounterCount;I++)
                              if ( send counter as class 2 )
                              DacWriteQueue( pEnv, devId, QUEUE_PRIORITY_LOW,
                                     (DacVariable) counter, i,
                                     counterevent, sizeof(COUNTERINPUTEVENT) );

                       p->Reason = DAC_DEVICE_REASON_INTERROGATE_STOP;

                       /* Write stop marker into queue */
                       DacWriteQueue( pEnv, devId, QUEUE_PRIORITY_LOW,
                              (DacVariable) device, off, p,
                              sizeof(DEVICEINPUTEVENT) );
                       break;

               case DAC_DEVICE_REASON_STOP_INTERROGATION :
```

```
                    /* GPT thinks interrogation is done */
                    /* GPT is sending ACCTERM */
        }
}
```