



Overview

This document describes how to pull counter data from the PointWare database. This application note includes an example program called "2005 Count.xls". The program is an Excel spreadsheet with a Visual Basic macro. The macro runs SQL queries to gather the counter data and place it in the spreadsheet. The demo assumes that 1 single beam people counter is assigned to a Point Server/Point Manager. A sample database is also provided. You will need to attach the database to the database server. You can download the example and database using this link: http://www.pointsix.com/download/AppNote_MiningCounterData.zip. Times stored with the counts in the database are in UTC. The counts are stored from 6/2/05 to 6/17/05 (EST). You will need to modify this line in the Visual Basic macro to work with your database server:

```
"ConnectionString = "Provider=SQLOLEDB.1;Password=pw;Persist Security Info=True;User ID=sa;Initial Catalog=PointWare;Data""
```

The application note gives example queries and then defines the fields in the database tables that are used.

General Notes:

1. HA10 refers to Point Managers and Point Servers
2. Sensors define attributes of the sensor and its I/O
3. Sample is a sensor I/O point
4. Node or HA10Node is a branch in the navigation tree
5. Most Counter Data are accumulated counts of a 24 bit counter. The maximum value is 16777215 before the counter will rollover.
6. The counter can be reset back to zero by pressing the service button on the sensor for 3 seconds.
7. To get counts per period you must take the difference of the counts during the period. Rollover and reset must be accounted for.
8. The following is list of reference indexes used to access particular counter types:

<u>Sensor Name</u>	<u>SDId</u>	<u>Maximum Count</u>
People Counter – Single	8	16777215
Directional Counter – CountA	34	16777215
Directional Counter – CountB	35	16777215

Obtaining Sensor Information

The example query will obtain a list of SamId's (list of indexes pointing to the Counter sensor I/O). The query will pull data for the single beam People Counter (SDId = 8). Note: The list of "SamId"s is used to query the counts

```
SELECT Id as Store, Name as Reader, SerialNo, SamId
FROM (tblSam INNER JOIN (tblSen INNER JOIN (tblHa10Cfg INNER JOIN tblHa10 ON
tblHa10.HId=tblHa10Cfg.HId) ON tblHa10Cfg.CId=tblSen.CId) ON tblSam.SenId=tblSen.SenId)
WHERE SDId = 8 AND Enabled = 1 ORDER BY SamId
```

<i>tblHa10</i>		
<u>Name</u>	<u>Type</u>	<u>Description</u>
Hid	Int	Reference
Nid	Int	Index into tblNode – used to create Path Name
Id	Varchar	Name of Point Manager
Enabled	Bit	Is setup active
Location	Varchar	
Description	Varchar	
Mac	Varchar	Ethernet MAC
UTCOffset	Real	Offset in days from UTC to Manager local time
DayLightSav	Bit	Observe Daylight Savings

<i>tblHa10Node</i>		
<u>Name</u>	<u>Type</u>	<u>Description</u>
Nid	Int	Index into tblNode – used to create Path Name
PNid	Int	Parent Node Id
NodeKind	Tinyint	Type of node
Name	Varchar	Name of the node

<i>tblHa10Cfg</i>		
<u>Name</u>	<u>Type</u>	<u>Description</u>
Cid	Int	Reference
Hid	Int	Index into tblHa10
Active	Bit	Is setup active

<i>tblSen</i>		
<u>Name</u>	<u>Type</u>	<u>Description</u>
SenId	Int	Reference
Cid	Int	Index into tblHA10Cfg
Name	Varchar	Name of the sensor
SerialNo	Varchar	Serial number of the sensor

<i>tblSam</i>		
<u>Name</u>	<u>Type</u>	<u>Description</u>
SamId	Int	Reference
SenId	Int	Index into tblSen
SDId	Int	Index into tblSenDef – sensor definition table

Obtaining Counter data

Simple View

```
SELECT tblDInteger.Data
FROM   tblRecord INNER JOIN
       tblDInteger ON tblRecord.RId = tblDInteger.RId
WHERE  (tblDInteger.SamId = " & current_samid & ") AND " &
       tblRecord.HistAt BETWEEN "' & start_date & "' AND "' & end_date & "'"
```

Note: *current_samid*, *start_date* and *end_date* are passed parameters.

Accounting for UTC, Daylight Savings and Local Time of the Point Server

```
SELECT tblDInteger.Data
FROM   tblRecord INNER JOIN
       tblDInteger ON tblRecord.RId = tblDInteger.RId
       INNER JOIN tblHA10Cfg ON tblHA10Cfg.CId = tblRecord.CId
       INNER JOIN tblHA10 ON tblHA10Cfg.HId = tblHA10.HId
WHERE  (tblDInteger.SamId = " & current_samid & ") AND
       (tblRecord.HistAt + (tblHA10.UTCOffset + CAST(tblHA10.DayLightSav AS REAL) / 24))
       BETWEEN "' & start_date & "' AND "' & end_date & "'"
```

Note: *current_samid*, *start_date* and *end_date* are passed parameters. *start_data* and *end_date* are the date ranges in local time in the format 'mm/dd/yyyy hh:nn:ss'.

	<i>tblDInteger</i>	
<u>Name</u>	<u>Type</u>	<u>Description</u>
SamId	Int	Index into tblSam
RId	Bigint	Index into tblRecord
Data	Int	Integer data – accumulated counts

	<i>tblRecord</i>	
<u>Name</u>	<u>Type</u>	<u>Description</u>
RId	Bigint	Reference
CId	Int	Index into tblHa10Cfg
HistAt	Datetime	Date/time of the history record. Store in UTC if UTCmode is set

Counter Status Information

	<i>tblDState</i>	
<u>Name</u>	<u>Type</u>	<u>Description</u>
SamId	Int	Index into tblSam
RId	Bigint	Index into tblRecord
Data	Int	State value. For Counters: 0 = Blocked 1 = Open 2 = Closed 3 = reserved

Activity Report

	<i>tblActivity</i>	
<u>Name</u>	<u>Type</u>	<u>Description</u>
ActId	Int	Reference
ActAt	Datetime	Date/time of activity (stored in local time of computer)
DevName	Varchar	Media Device Name
HA10Id	Int	Index into tblHA10 (NULL if no HA10)
ActionId	Tinyint	Index into tblActionDef
Msg	Varchar	Message describing activity or error
aError	Bit	Indicates if activity is an error message

	<i>tblActionDef</i>	
<u>Name</u>	<u>Type</u>	<u>Description</u>
ActionID	Tinyint	Reference
Description	Varchar	Action Description