OSIsoft.

vCampus Live! 2013

Custom Applications for PI Data Transfer, Monitoring and Reporting

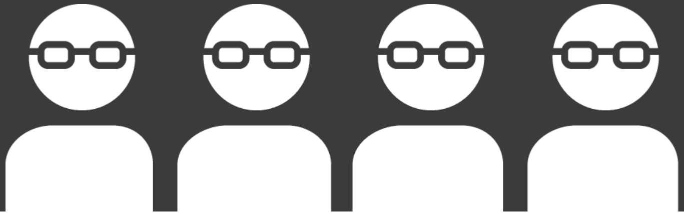
Presented by **Thomas L. Roach, P.E.**

BP Products North America, Inc.

OSIsoft®

vCampus Live! 2013

WHERE PI GEEKS MEET



BP Whiting Business Unit (“WBU”) PI System

- The PI System has been in use at Whiting since 1989.
- Primary site process historian, with over 117,000 points currently.
- Mainly Honeywell and Emerson control systems.
- Hardware & software platforms
  - Legacy DEC VAX© mainframes running VMS©.
  - Microsoft Windows© Servers/Workstations.

Why? Custom Applications Why?

- Advantages
  - get “exactly” what you want (or at least what you asked for...).
  - “home-grown” is readily/timely customizable.

- Disadvantages
  - can be costly and time-consuming to create.
  - sometimes difficult to support and maintain.

WBU Custom PI Data Applications

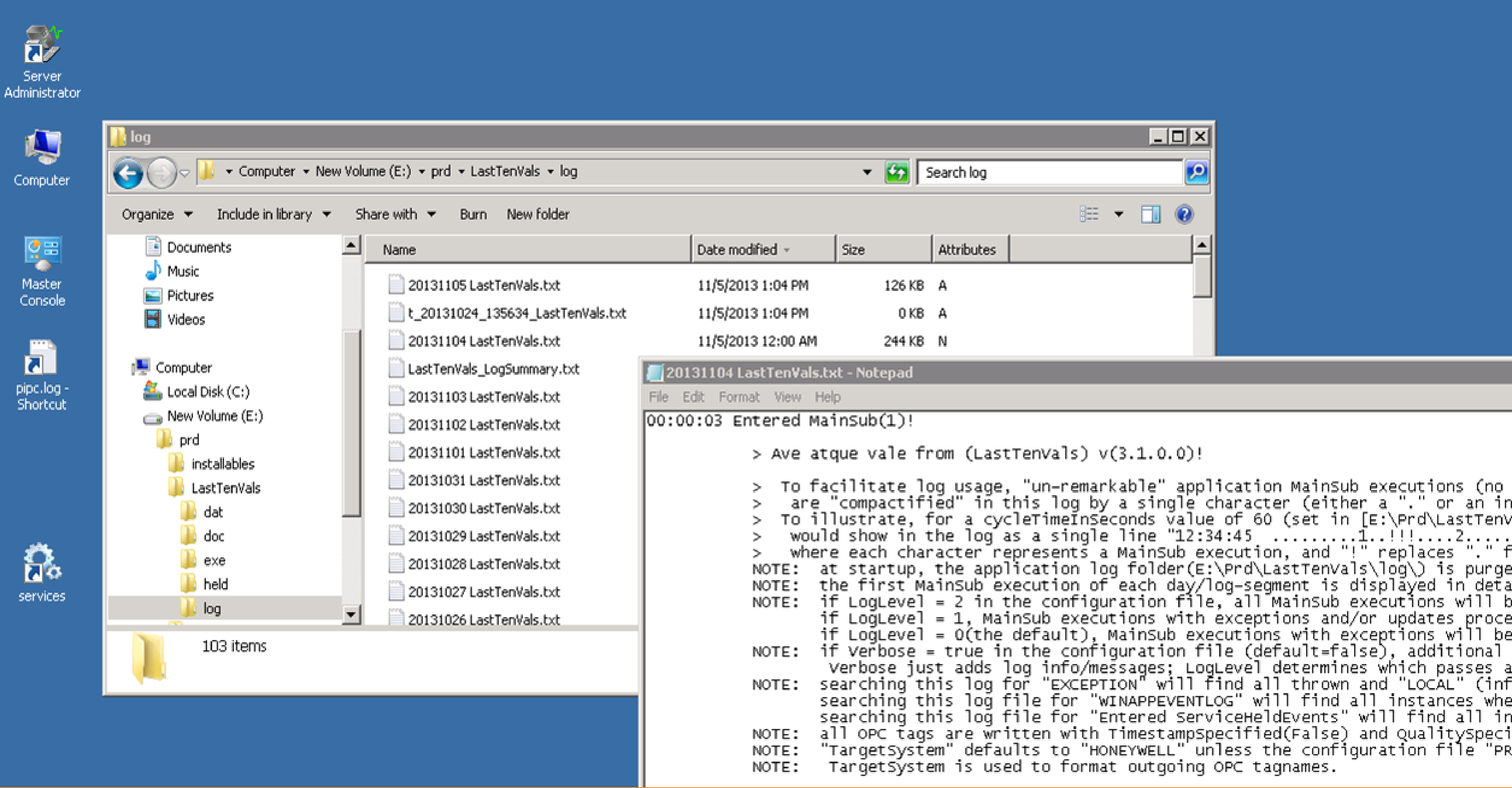
- 12 applications, 33 running instances, each requested to perform necessary functions that are “outside of standard PI capabilities”.
- Created using Microsoft Visual Studio VB.NET with the OSIsoft PI SDK and OpcNetApi.
- We’ve migrated some legacy applications, and developed some new ones.
- Mainly Microsoft Windows Services, with some Console Applications, and some Excel Automation.
- For data transfer, with automated review, analysis and reporting.

Application Types

1. **Simple** data transfer; fixed interval, PI->OPC.
2. More **complex** data transfer.
3. "**Event-driven**" applications that transfer data and generate output files/records in response to trigger events.
4. "**Monitoring and Reporting**"; analyze operational data, email alerts & daily reports.
5. **Other**; Excel automation, SQL data.

Common Features – Installation Packaging

- Visual Studio Installer.
- Installation creates & populates a folder structure for application support.
- PI Network Manager dependency added for Services.
- Version numbering.
- Full installation v executable replacement.
- Working class “kernel” for multiple instances.
- Updating to Visual Studio 2013, InstallShield.



Name	Date modified	Size	Attributes
20131105 LastTenVals.txt	11/5/2013 1:04 PM	126 KB	A
t_20131024_135634_LastTenVals.txt	11/5/2013 1:04 PM	0 KB	A
20131104 LastTenVals.txt	11/5/2013 12:00 AM	244 KB	N
LastTenVals_LogSummary.txt			
20131103 LastTenVals.txt			
20131102 LastTenVals.txt			
20131101 LastTenVals.txt			
20131031 LastTenVals.txt			
20131030 LastTenVals.txt			
20131029 LastTenVals.txt			
20131028 LastTenVals.txt			
20131027 LastTenVals.txt			
20131026 LastTenVals.txt			

```
00:00:03 Entered MainSub(1)!

           > Ave atque vale from (LastTenVals) v(3.1.0.0)!

           >  To facilitate log usage, "un-remarkable" application MainSub executions (no
           >   are "compactified" in this log by a single character (either a "." or an in
           >  To illustrate, for a cycleTimeInSeconds value of 60 (set in [E:\Prd\LastTenV
           >   would show in the log as a single line "12:34:45  .........1..!!!....2.....
           >   where each character represents a MainSub execution, and "!" replaces "." f
           NOTE:  at startup, the application log folder(E:\Prd\LastTenVals\log\) is purge
           NOTE:  the first MainSub execution of each day/log-segment is displayed in deta
           NOTE:  if LogLevel = 2 in the configuration file, all MainSub executions will b
                  if LogLevel = 1, MainSub executions with exceptions and/or updates proce
                  if LogLevel = 0(the default), MainSub executions with exceptions will be
           NOTE:  if Verbose = true in the configuration file (default=false), additional
                   Verbose just adds log info/messages; LogLevel determines which passes a
           NOTE:  searching this log for "EXCEPTION" will find all thrown and "LOCAL" (inf
                  searching this log file for "WINAPPEVENTLOG" will find all instances whe
                  searching this log file for "Entered ServiceHeldEvents" will find all in
           NOTE:  all OPC tags are written with TimestampSpecified(False) and QualitySpeci
           NOTE:  "TargetSystem" defaults to "HONEYWELL" unless the configuration file "PR
           NOTE:   TargetSystem is used to format outgoing OPC tagnames.
```

Common Features – Configuration File

- Specifies values that may need to be changed but without rebuilding and re-installing the application.
- .csv file format.
- CYCLE_TIME_IN_SECONDS, PI_SERVER_NAME, READ_ONLY_OPERATION, etc.
- Transfer tag lists (... or in separate file).
- Configuration file template packaged for initial installation, but so as to NOT OVER-WRITE the existing active file on a re-install.

Common Features – Status/Error Log File

- Generated daily.
- Header with basic operational and configuration details, the running version#, and datetime of the last restart.
- Un-eventful passes are “compactified” by first writing to a temporary-primary log file.
- Operational-summary footer for each log-segment, also written to 20-deep LogSummary file.

Log File “Compactification”

```
15:37:35  .........1.........2.........3.........4.........5.........6
16:37:35  .........1.........2.........3.........4.........5.........6
17:37:35  .........1.........2.........3.........4.........5.........6
18:37:35  .........1.........2.........3.....

19:12:35  Entered MainSub(1153)!
19:12:35   CHECKING for input files(MyInputFileSpec)!
19:12:35   FOUND(1) input-candidate file!
19:12:35
19:12:35   >inputFile(1) = (MyInputFilePath), for BlenderNumber(3)!
19:12:35   ...MAINLINE PROCESSING:
19:12:35   Entered ReadInputFile(actual)!
19:12:35    Read(26) lines from file(MyInputFilePath)!
19:12:35    inputLine(0) - [<<BLEND MANAGEMENT>>]!
19:12:35    inputLine(1) - []!
```

Common Operating Features

- Startup behavior:
  - minimal OnStart() initialization for quick Service startup.
  - run the first “working” pass promptly and detailed to the status log for operational verification.
- Timed execution with re-entrancy provisions.
- Designed to “ride out” exceptions and PI/OPC connection interruptions.
- Issue email alerts for recurring/critical exceptions, with RTN’s.
- Execute orderly-shutdown tasks for application/server shutdowns.
- Save/GetSetting used to carry critical values over between application shutdowns and subsequent startups.

Common Features – PI Connectivity

- PI Servername alias look-up to simplify startup deployment.
- PI connectivity is established at application startup, then tested each pass and re-established when needed.
- PISDK.Refresh(PointList) executed each pass to pick up attribute changes (for example, digitalset).

Common Features – OPC Connectivity

- OPC connectivity is established at application startup, then verified each pass.
- OPC is programmatically disconnected in response to connection/transfer problems, then automatically re-establishes on the next pass.
- Primary/secondary OPC Server swap for “automatic” fail-over.
- Measure & log OPC write times, treat as an exception when > configurable threshold.

Common Features – Exception Alerts

- Using SolarWinds ipMonitor©:
  - Server or Service not running.
  - app-exception Windows Application Event Log entries.
- ReturnToNormal (RTN) notification issued once “full”/meaningful operation has been verified.
- Exele TopView© also used, to alert when application-output PI values stop updating.
- Alert and notification emails are sent by some applications via an intermediate custom utility, by others directly via System.Net.Mail.SmtpClient.

The joy of SmtpClient emailing...

```
Imports System.Net.Mail


Using mailer As New SmtpClient(“MySmtpServerIP")
    mailer.Timeout = 15000
    mailer.Send(“sender@my.mail.com",
                “recipient@other.mail.com",
                “MyEmailSubject",
                “MyEmailBody")
End Using
```

Just a few lines of code for an email! The recipient string can contain a list of addresses separated by semicolon(s). The email body can be plain text or HTML.

“PItrans”

- Simple data transfer application.
- Fifteen running instances, with one “double”.
- Replaces DEC FORTRAN and C++ legacy apps.
- Designed to regularly transfer value, timestamp and quality data from PI -> OPC points.
- RecordedValuesByCount(Now(),1) with .dReverse and .btInside.
- Selected points are readback-verified.

PItrans – build output taglist, write to OPC

```
Public OPCItems() As Opc.Da.ItemValue
Public iOPCItems As Integer = 0
..

OPCItems(iOPCItems) = New Opc.Da.ItemValue

OPCItems(iOPCItems).ItemName = outTag(i).ValueTagName
OPCItems(iOPCItems).Value = outTag(i).Value
OPCItems(iOPCItems).Timestamp =
       outTag(i).TimeStamp.LocalDate.ToString
OPCItems(iOPCItems).Quality = Opc.Da.Quality.Good/Bad
..

Dim irs() As IdentifiedResult = OPCsvr.Write(ptList)
```

“LastTenVals”

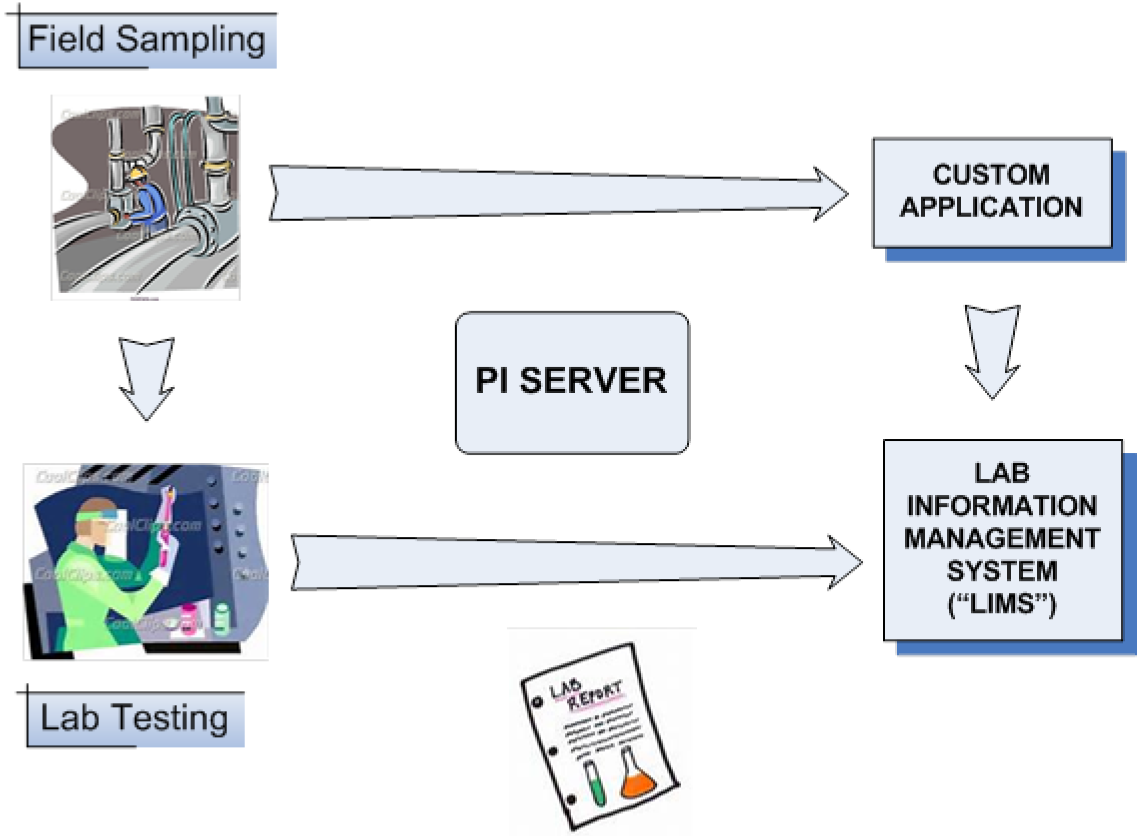
- More complex data transfer, 8 installed instances, 5 running on a single Server for ~year.
- Replaces C++ legacy applications.
- Designed to push the last ten chronological lab test values to an operator control board comparison display.
- PISDK.PointList.Data.EventPipe.Count/TakeAll.
- RecordedValuesByCount(Now, 10,.dReverse,.btInside) built into outgoing Opc.Da.ItemValue array.
- Selected points are readback-verified.
- Auto/manual heldEventTagnames ArrayList.
- Picks held or random point for startup exercise.

LastTenVals – ServiceTheEventPipe() excerpt

```
For Each peo As PIEventObject In evPipe.TakeAll
    tagname = peo.EventData.PIPoint.Name.Trim.ToUpper

    If PushLastTenPiValuesToOpcPointList(tagname) = True
       AndAlso WritePointListToOPC(OPCoutgoingPointList) = True
    Then
    '   operation succeeded; remove tagname from held events arraylist.
        ServiceTheEventPipe = ServiceTheEventPipe And True
        heldEventTagnames.Remove(tagname)
    Else
    '   operation failed; add tatgname to held events arraylist.
        ServiceTheEventPipe = False
        If heldEventTagnames.Contains(tagname) = False Then
            heldEventTagnames.Add(tagname)
        End If
    End If
Next
```

Blend and Shipment Sample Data Management



“ShipCap” – capture shipment events

- “Event-driven” application.
- Designed to facilitate LAB INFORMATION MANAGEMENT SYSTEM (“LIMS”) generation of sample records and printed labels for pipeline & barge shipments.
- Watches OPC status flags signaling pipeline/barge shipment events (start, mid, end, reprint).
- Generates output record for SFTP transfer to LIMS, which generates the sample record and prints lab-sample tags.
- Prints informational report to the Lab Supervisor’s office.

ShipCap – Reading a status flag

```
Private Function StatusFlagFromLCN(...) As Single

      Dim OPCdaItem(0) As Opc.Da.Item
          OPCdaItem(0).ItemName = pipelines(pIndex).LCNpoint & "." &
                                      ReportTypes(rIndex).StatusFlagFieldName

     '   read target OPC point
      For Each itr As ItemValueResult In OPCsvr.Read(OPCdaItem)
          retValStr = itr.Value
          If Not itr.Quality.ToString.ToUpper.Contains("GOOD")
             Or Not itr.ResultID.Succeeded = True
          Then
             '   read was problematic; trigger DisConnectOPC() for next-pass reconnection.
              ThereWasAnOPCexceptionThisPass = True
           End If
      Next

      '   assign return value
      Single.TryParse(retValStr, StatusFlagFromLCN)
```

“BlendCap” - capture blend events

- “Event-driven” application.
- Designed to facilitate LIMS generation of sample records and printed labels for gasoline and distillate blend sampling events. Sets up synchronizing timestamp for operator display of lab v analyzer results.
- Shadows OPC “sample-capture” sequencing flags to PI tags.
- Retrieves blend information and analyzer data.
- Writes PI data and LIMS records/printouts using the sample-event timestamp.
- When lab tests are run and LIMS sends the results to PI, they are timestamped identically to the analyzer values for comparison display (see “SALAD” following...).

BlendCap – Writing a sequencing flag to PI

```
If piSvr Is Nothing Or piSvr.Connected = False Then piSvr.Open()
Dim ptval As PIPoint = piSvr.PIPoints.Item(tagName), pvs As PIValues

For i As Integer = 1 To 3      '   make up to three attempts
   Try
       ptval.Data.UpdateValue(tagValue, Now, DataMergeConstants.dmReplaceDuplicates)
       pvs = piSvr.PIPoints(ptval.Name).Data.RecordedValuesByCount(sTime + halfSec,
              1, DirectionConstants.dReverse, BoundaryTypeConstants.btInside)
       If pvs.Count > 0 Then
           If Single.TryParse(pvs(1).Value, tmpInt) = True AndAlso tmpInt = tagValue Then
               WriteSequenceFlagToPI = True
               Exit For
           Else
               numFailedAttempts += 1      '   read/write values don't match
           End If
       Else
           numFailedAttempts += 1      '   couldn't read back the written value
       End If
   Catch ex As Exception
       numFailedAttempts += 1      '   some error occurred
   End Try
Next
```

“SALAD” - match lab results to analyzer values

- “Event-drive” application.
- “Synchronized Analyzer Lab Analyses Distribution”.
- Lets the operator see side-by-side field-analyzer v lab-test results for the last ten samples taken for each gasoline & distillate blender.
- Creates a set of nested structures that holds LABRESULT and ANALYZERDATA PI point values.
- EventPipe catches updates for Lab test results and analyzer data PI points.
- New LABRESULT data values are saved into the existing structure that has the same timestamp (courtesy of BlendCap).
- New ANALYZERDATA values are saved into the matching-timestamp existing structure if one exists; new timestamps roll vector(10) out and create a new vector(1).

SALAD – data structures' initialization

```
For i As Integer = 0 To numBlenders    ‘  each blender has a LABDATA and ANALYZERDATA struct
    For j As Integer = 0 To 50         ‘  each consisting of 50 parameters of interest
        ANALYZERDATA(i).PV(j).tagName = "."
        ANALYZERDATA(i).PV(j).description = "."
        LABDATA(i).PV(j).tagName = "."
        LABDATA(i).PV(j).description = "."
        For k As Integer = 0 To 10     ‘  for each of which 10-deep values are stored
            ANALYZERDATA(i).PV(j).last10values(k) = defTag
            LABDATA(i).PV(j).last10values(k) = defTag
        Next
    Next
 Next
```

"EOBLIMS" - end-of-blend processing

- "Event-driven" application.
- Designed to detect and process end-of-blend events, to facilitate lab sample testing and detect any actual v target discrepancies.
- Watches for end-of-blend data files FTP'd from blend management system via intermediate application.
- Reads end-of-blend data, creates and forwards a blend-sample-login file to LIMS, generates a report to the Lab Supervisor's printer.
- Detects and issues notifications for ACTual v TARget discrepancies; missing data, multiple values, value mis-matches.

EOBLIMS – input file structure/parsing

```
Public Structure InputData
    Dim BlendedQuantity As Single, BlendedQuantityIsValid As Boolean
    Dim BlenderName As String, BlenderNameIsValid As Boolean
    Dim BlendNumber As Integer, BlendNumberIsValid As Boolean
    Dim BlendStart As Date, BlendStartIsValid As Boolean
    Dim BlendStop As Date, BlendStopIsValid As Boolean
    Dim BlendTank As String, BlendTankIsValid As Boolean
    Dim HeaderVolume As Single, HeaderVolumeIsValid As Boolean
    Dim HeelVolume As Single, HeelVolumeIsValid As Boolean
    Dim Product As String, ProductIsValid As Boolean
    Dim Recipe As String, RecipeIsValid As Boolean
    Dim AllDataFieldsAreOK As Boolean, DataFieldsForTARvACTcomparisonAreOK As Boolean
    Dim whyTarAndActFilesDontMatch As String
End Structure

'   Parse out the heel volume
If Single.TryParse(inputlines(23 - lineNumOffset).Replace("HEEL_QTY,", ""),iData.HeelVolume) = False Then
    msg = "FAILED to parse HeelVolume from(" & inputlines(23 - lineNumOffset) & ")!"
    sLog(t_lfp, Format(Now, "HH:mm:ss  ") & tab & sp & msg)
Else
    iData.HeelVolumeIsValid = True
End If
```

“EnviroMon” - environmental monitoring

- “Monitoring and Reporting” application.
- Designed to monitor environmental data in 7 performance areas, including operating unit CO, SO_2 and NOx emissions, particulates, and fire-pump-engine running time.
- Checks flare-monitoring-data for completeness & validity, reports any gaps over 15 minutes.
- Monitoring for each area can be separately enabled or set to read-only mode for test/development.
- Issues 05:00 prior-day performance summary reports, with High importance (“!”) early-warning alert notifications for actionable conditions.

EnviroMon – daily data averaging

```
' grab values for the period of interest
Dim pivs As PIValues = piSvr.PIPoints(tagName).Data.RecordedValues(sDateTime, eDateTime,.btInterp)
If pivs.Count > 0 Then
    ' check for bad values
    For i As Integer = 1 To pivs.Count
        If pivs(i).Value.GetType.IsCOMObject = True OrElse pivs(i).IsGood = False
           OrElse Date.TryParse(pivs(i).TimeStamp.LocalDate.ToString, tmpDate) = False Then
               numBadPiValues += 1
        End If
    Next
    numUsablePiValues = pivs.Count – numBadPiValues
    ' calculate the average if there are usable values
    If numUsablePiValues > 0 Then
        Dim piv As PIValue = pivs.Summary(sDateTime,eDateTime,.astAverage,.cbTimeWeightedContinuous)
        If piv.Value.GetType.IsCOMObject = False Then
            PiDataAverage = piv
        Else ...
        End If
    End If
End If
```

EnviroMon – example report excerpt

```
From:  [hard-coded sender]
Sent: Monday, August 19, 2013 5:01 AM
To:  [configurable recipients]
Subject: FCU 500 NOX Monitoring Report for (Sun 18-Aug-2013)

NOX Averages [ppm] for (Sun 18-Aug-2013):

  Daily Average ............. (#.#)  OK!  (< #.#)
  7-Day Rolling Average ..... (#.#)  OK!  (< #.#)
  365-Day Rolling Average ... (#.#)  CURRENTLY NOT EVALUATED for alerting purposes

NOTE:  reported data is calculated from archived values for the following PI tags:

  FCU500 Daily Average ................... ABCXYZ123
  FCU500 7-Day Rolling Average ........... ABCXYZ123
  FCU500 365-Day Rolling Average ......... ABCXYZ123
  FCU500 CEM O2 .......................... ABCXYZ123
  FCU500 5-1 Loader Rate ................. ABCXYZ123
  FCU500 5-2 Loader Rate ................. ABCXYZ123
  FCU500 Feed Nitrogen ................... ABCXYZ123
  FCU500 Ammonia Injection ............... ABCXYZ123
```

“FlareCameraMon” - flare DVR system interface

- “Monitoring and Reporting” application.
- Designed to assure USEPA-mandated flare video “long term recording” compliance.
- Set up to monitor PI status tags for 8 flare camera systems.
- Two redundant parallel application instances run on separate servers communicating with separate redundant DVR systems.
- Each pass is constrained to execute at :00 seconds for “simultaneous” operation.
- RS-232 serial interface to DVR systems, reads status information for alerting purposes, writes control commands to start/stop long-term recording.

FlareCameraMon – serial input excerpt

```
If serialPort.BytesToRead > 0 Then serialPortInputString = ReadSerialPort.Trim.ToUpper

Public Function ReadSerialPort() As String
    Dim rByte As Integer = 0, rBs(1) As Integer, nBytesRead As Integer = 0
    rBs(0) = 0

    While serialPort.BytesToRead > 0   ‘ each received byte is a decimal-integer-encoded ASCII character.
        rByte = serialPort.ReadByte
        nBytesRead += 1
        ReDim Preserve rBs(nBytesRead)
        rBs(nBytesRead) = rByte          ‘ add to the received-byte array.
    End While

    For i As Integer = 1 To nBytesRead   ‘ return received bytes as a string.
        If rBs(i) < 32 Or rBs(i) > 127 Then
            ReadSerialPort += "[" & rBs(i).ToString & "]"
        Else
            ReadSerialPort += Chr(rBs(i).ToString)
        End If
    Next
```

“Other” applications - Excel Automation

- Automating Excel + PI DataLink capabilities.
  1) Console application(1) opens and re-calculates a refinery-utilization spreadsheet, scrapes and stores KPI’s to PI for management scorecard.
  2) Console application(2) opens spreadsheet alarm reports, scrapes & stores KPI’s to PI for management scorecard.
  3) Windows Service back-calculates-to-convergence and writes wet-bulb temperature values to PI, replacing Excel.GoalSeek manual spreadsheet.
- Caveats:
  – Dev/Use version conflicts: Excel Interop/library packaging, late binding of Excel objects.
  – Management of multiple Excel instances.

Other applications - “pipeCorrDataConApp”

- Windows Console Application designed to retrieve SQL data for 145 pipe thickness sensors and write it to PI for ready correlation with other information.
- Creates and fills a local dataset/table to minimize SQL connection time and resource consumption.
- Succinct SQL interface enabled by the .NET System.Data.SqlClient namespaces.

pipeCorrDataConApp – SQL data retrieval

```
Imports System.Data.SqlClient
   Public tblOfThicknessData As DataTable = New DataTable("thicknesses")
   Dim bldr As New SqlConnectionStringBuilder, bldr("Data Source") = “MySqlServer“ ...
    Using cnn As New SqlConnection(bldr.ConnectionString)
        Using cmd As New SqlCommand()
            cmd.Connection = cnn
            Using ta As New SqlClient.SqlDataAdapter(cmd)
                cmd.CommandText = “MySqlServer“
                ta.Fill(tbl_thickness_data)
            End Using
        End Using
        If cnn.State = ConnectionState.Open Then cnn.Close()
    End Using
```

Anomalies Encountered – OPC writes

Using OpcNetApi, only encountered with ControlSystemVendorB.

1) Custom client memory leak:
   - Opc.Da.ItemValue array scope v persistent group creation.
   - Known problem, OPCRemote update to fix.

2) E_UNKNOWN_ITEM_NAME and E_FAIL on first write attempts.
   - non-fatal, but evidencing larger problems?
   - investigation ongoing.

CONCLUSION

- Business case for custom and “shrink-wrap” programmatics.
- The available tools will continue becoming ever more powerful (and satisfying/FUN to use!).
- Balancing the cost of keeping current v falling behind.
  - time & effort to upgrade v growing vulnerability.
  - OS/patching, application and utilities compatibility.
  - maintenance and support legacy.

Thomas L. Roach, P.E.

Process Computing Analyst

BP Products North America, Inc.

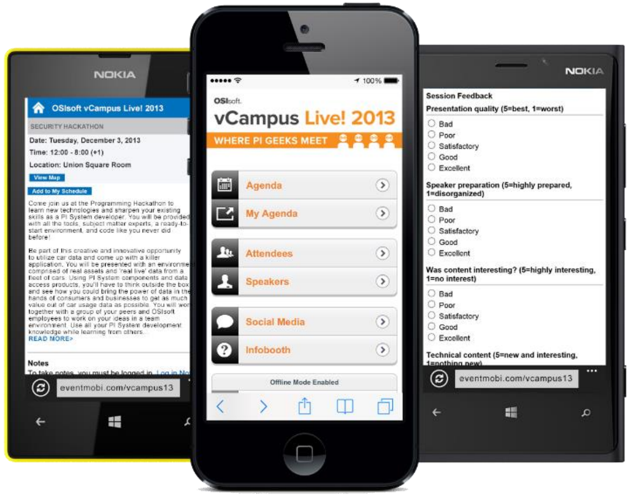
Whiting Business Unit

thomas.roach@bp.com

Please don't forget to…

Complete the online survey for this session

eventmobi.com/vcampus13



SECURITY HACKATHON

Date: Tuesday, December 3, 2013

Time: 12:00 - 8:00 (+1)

Location: Union Square Room

Come join us at the Programming Hackathon to learn new technologies and sharpen your existing skills as a PI System developer. You will be provided with all the tools, subject matter experts, a ready-to-start environment, and code like you never did before!

Be part of this creative and innovative opportunity to utilize car data and come up with a killer application. You will be presented with an environme comprised of real assets and 'real live' data from a fleet of cars. Using PI System components and data access products, you'll have to think outside the box and see how you could bring the power of data in the hands of consumers and businesses to get as much value out of car usage data as possible. You will wor together with a group of your peers and OSIsoft employees to work on your ideas in a team environment. Use all your PI System development knowledge while learning from others.

Notes

To take notes, you must be logged in. Log in No

Session Feedback

Presentation quality (5=best, 1=worst)

- ○ Bad
- ○ Poor
- ○ Satisfactory
- ○ Good
- ○ Excellent

Speaker preparation (5=highly prepared, 1=disorganized)

- ○ Bad
- ○ Poor
- ○ Satisfactory
- ○ Good
- ○ Excellent

Was content interesting? (5=highly interesting, 1=no interest)

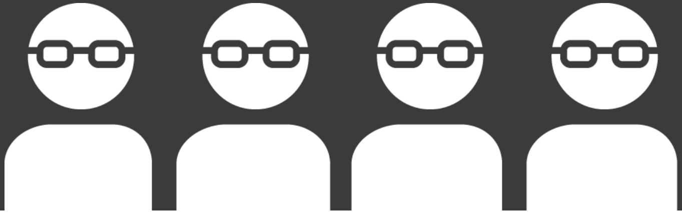
- ○ Bad
- ○ Poor
- ○ Satisfactory
- ○ Good
- ○ Excellent

Technical content (5=new and interesting, 1=nothing new)

Share with your friends

#VCL13

THANK YOU



Brought to you by OSIsoft.