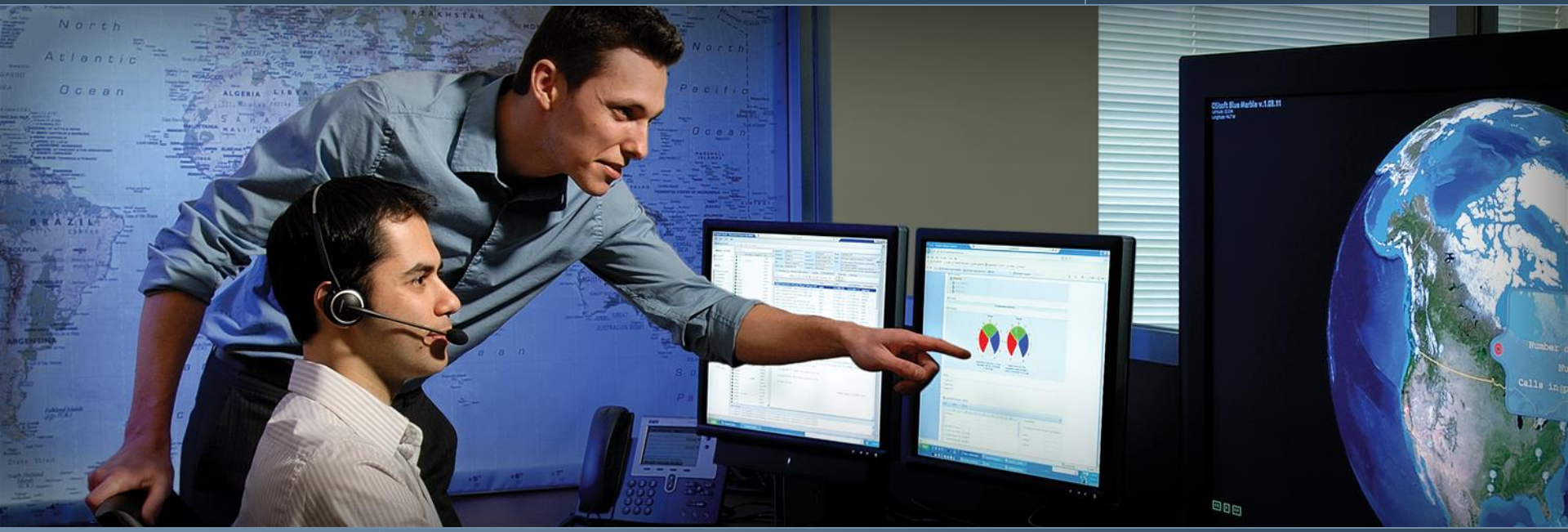
Regional Seminar Series
Raleigh



PI Interfaces

Julie Zeilenga
Software Engineering Group Leader
OSIsoft, LLC.

October 27, 2009

Empowering Business in Real Time.

Agenda

- Interfaces Used in Critical Facilities, Data Centers and IT
- AMI Interfaces to Head End Systems
- New Interface Framework to Replace UniInt
- Widely Used OSIsoft Interfaces
- Batch Interfaces Developed with New Batch Framework – Todd Brown

Facilities, Data Center and IT Interfaces

- OPC, RDBMS
- Modbus, Perfmon
- SNMP, Windows Event Log, Ping
- SNMP Traps, TCP Response, Syslog
- IP Flow (NetFlow)
- *IPMI
- BACnet

BACnet Overview

- BACnet stands for building automation and control network
- BACnet provides a common protocol to interface with building automation and control systems
- BACnet is supported by many leading building control systems vendors:
  - Automated Logic Corporation
  - Johnson Controls Incorporated
  - Siemens Building Technologies
  - Tridium

BACnet Overview (Cont.)

- The BACnet protocol defines a number of services that are used to communicate between building devices
- Protocol services include: *Who-Is*, *I-Am*, *Who-Has*, *I-Have*, which are used for Device and Object discovery
- The PI BACnet interface supports **BACnet/IP** over Ethernet
- The interface supports a BACnet Query Tool

AMI Head End Interfaces

- Advanced Metering Infrastructure
  - Measurement of data
  - Data collection – lots of it!
  - Data Analysis
  - Two way communication with the meter
  - Commercial and residential meters
  - Electric, gas, and water

PI AMI Interface Functionality

- Meter connect/disconnect
- On demand reads (register read)
- Scheduled reads (intervals over a period)
- Pings
- Last gasp event message
- Meter asset information transferred to AF
- VEE – Validation , editing, estimation

What Have We Learned with AMI?

- Level playing field when we initially got involved
- Current interface framework won't work!
  - Not scalable enough
  - No easy integration with AF for building assets
  - No easy integration with business systems
- Other parts of our system not scalable/efficient enough

AMI Interfaces in Development

- Silver Spring Networks UtilityIQ (UIQ)
- Elster EnergyAccess Metering Automation Server (MAS)
- Grid Net PolicyNet
- Trilliant Unity
- *Itron OpenWay

New Interface Framework in Development

- UniInt developed initially in 1992
  - Common code to exchange information with the PI Server
- New Interface Framework
  - Real time data scalability
  - Exchange of asset information with AF
  - Built in point creation
- Initial Interfaces
  - OPC UA Client
  - New PIToPI Interface
  - AMI Head End Interfaces

Status of Widely Used PI Interfaces

- UniInt
  - Health tag information sent to performance tags
  - Addition of severity codes in messages for mPI
  - Phase III failover, secure socket connection between interface nodes
- OPC DA Interface
  - Performance improvements
- PIToPI Interface
- RDBMS Interface
  - Failover and history recovery
- *UFL
  - Performance improvements and failover
- *Batch Interfaces on the Batch Framework

Batch Interface Framework Overview

What is the Batch Interface Framework?

The *Batch Interface Framework (BIF)* is a new core library of functions for batch/event frame interfaces that is similar to UniInt.

- **1 -** Increase Interface Robustness
- **2 -** Treat source system data as the source & no more manual admin cleaning of data due to outages
- **3 -** Store Batch Events in a way that aligns with customer display and reporting use cases
- **4 -** Better handle small batch recipes in flexible manufacturing environments

  One common view of automated recipe steps and manual electronic work instructions
- **5 -** Increased demand for Batch Interfaces

Customer Goals

1
2
3
4
5

- **1 -** Decrease Support Calls
- **2 -** Self healing design

  Decrease Support Calls
- **3 -** Identify a Flexible Mechanism for Storing Batch Events

  Increase the Value of Batch Event Data
- **4 -** Provide one view of PI Batch information from the customer's standpoint
- **5 -** Quicker time to market for new Batch Interfaces

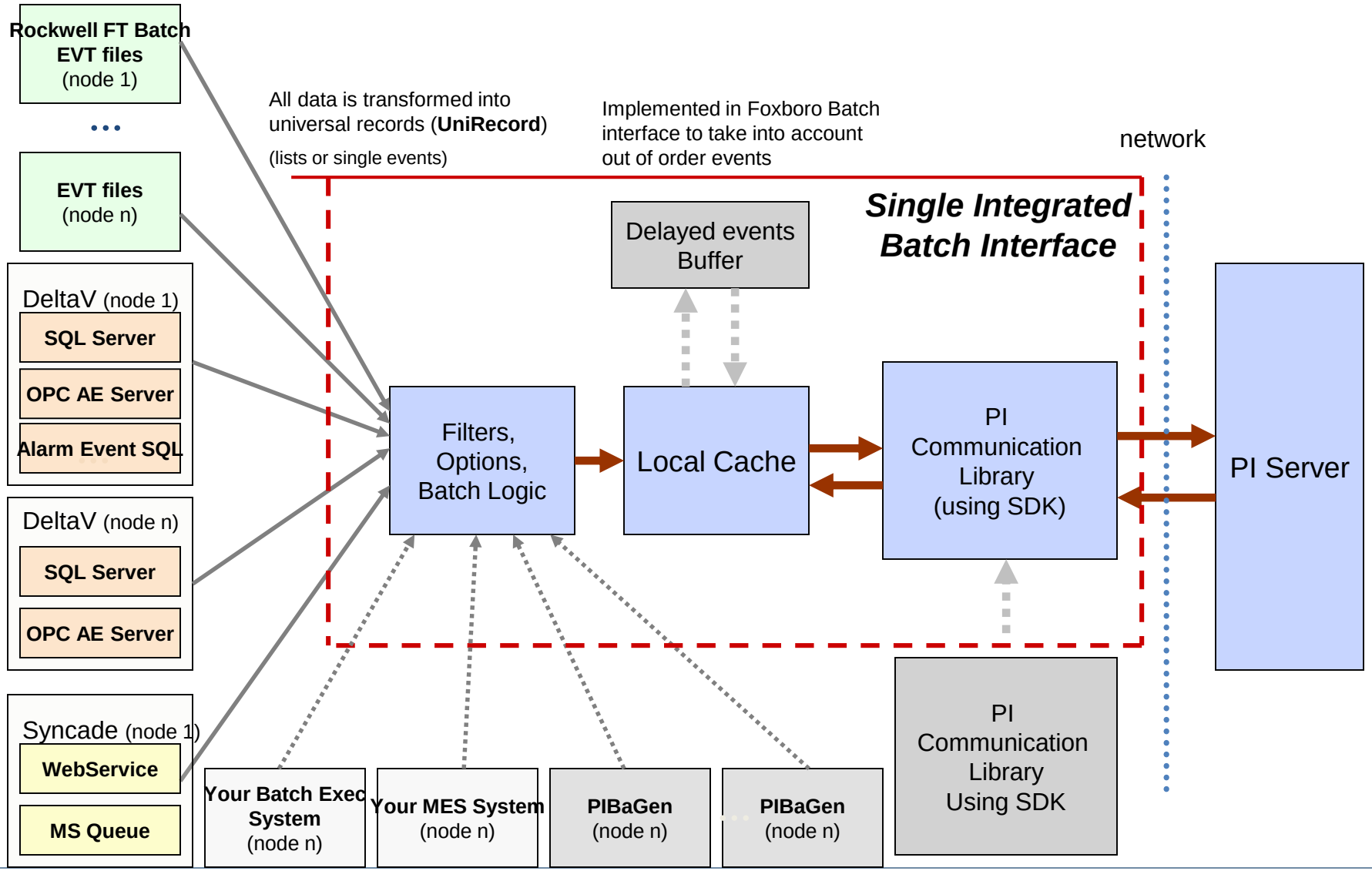
  Reduce Support Calls

  Reduce Code Maintenance

OSIsoft Goals

Value now. Value over time.

Batch Interface Framework Architecture



Value now. Value over time.

BIF Released Batch Interfaces (Aug 2009)

Supported Vendor Versions and Connectivity Methods

Emerson DeltaV Batch Interface (EMDVB)

- DeltaV v8.4: .EVT Files
- DeltaV v9.3: DeltaV Batch Historian or .EVT Files + SQL Alarm & Events Historian
- DeltaV v10.3: DeltaV OPC Alarm & Events Server + DeltaV Batch Historian + SQL Alarm & Events Historian or .EVT Files

Emerson Syncade Batch Interface (EMBCS)

- Syncade v4.0.1: Microsoft Message Queue / Web Service

Emerson DeltaV Syncade Batch Interface (EMDVBCS)

- Same as above (combination)

Rockwell FactoryTalk Batch Interface (PIFTBInt)

- RSBatch: .EVT Files
- FactoryTalk Batch: .EVT Files

ABB 800xA Batch Interface (ABB800xA)

- ABB 800xA: Oracle9i Release 2 Provider for OLEDB 9.2.0.7.0

Value now. Value over time.

Batch Interface Framework Functionality Overview

Tag Templates Functionality Overview

- Interface .bat file command line parameter /inifile=<full path filename> defines the location for the template file.
- Tag Templates allow defining structures for tag name, tag data type, tag value, unit alias name, phase module alias name, engineering units and descriptor properties.

Tag[index**].Name =** Name structure (with embedded triggering Event Type or Event Type Mask or Expression)
Tag[index**].Descriptor =** value structure as free text (default: **blank**)
Tag[index**].EngUnits =** value structure as free text (default: **blank**)
Tag[index**].Value =** Event value structure as free text
Tag[index**].Type = string/integer/float**
Tag[index**].UnitAlias = <**optional sub unit module path |> unit tag alias name structure (default: alias name as **.Name** and alias module path as **[Unit]**)
Tag[index**].PhaseAlias = <**optional sub phasemodule path |> phase module tag alias name structure (default: alias name as **.Name** and alias module path as **[PhaseModule]**)
Tag[index**].Trigger =** Event Type or Event Type mask or Expression
Tag[index**].Translate = true/false** (default: **false**)

- Multiple tag templates can be triggered by the same source event, and a single template can be triggered by multiple source events.
- Multiple tag templates are capable of writing to the same PI tag (if the .Name attribute of the tag templates resolves to the same PI tag name). This is useful when you want different values to be written to the same PI tag dependent on the trigger for each. Example:
  - Unit Batch Start Event TRIGGER, write a value of *Active*
  - Unit Batch Stop Event TRIGGER, write a value of *Inactive*
- Wildcards are supported in templates
- Placeholders are supported (next slide)

Wildcard	Description
#	single digit numerical value (0-9)
@	single alpha character (a-z, A-Z)
?	any single valid symbol
*	An array of valid symbols
!	repeat previous mask symbol

Tag Templates Functionality
Placeholders

Tag Templates can contain placeholders which are vendor / system specific

Emerson DeltaV Batch Historian Placeholders

- [TIME], [BATCHID], [PROCEDURE], [UNITPROCEDURE], [OPERATION], [PHASE], [DESCRIPT], [EVENT] or [PARAMETER], [PVAL] or [VALUE], [EU], [AREA], [PROCESSCELL], [UNIT], [PHASEMODULE], [USERID] or [USER], [UNIQUEID]

Emerson DeltaV Alarms & Events Placeholders

- [TIME], [EVENT], [CATEGORY], [NODE], [AREA], [PROCESSCELL], [UNIT], [MODULE], [MODULEDESC], [ATTRIBUTE], [STATE], [LEVEL], [DESC1], [DESC2]

Emerson Syncade Placeholders

- [TIME], [BATCHID], [PROCEDURE], [UNITPROCEDURE], [OPERATION], [PHASE], [DESCRIPT], [PARAMETER], [VALUE], [USER], [AREA], [PROCESSCELL], [UNIT], [UNIQUEID],[SET],[HIGH],[LOW]

Rockwell FactoryTalk Batch Placeholders

- [TIME], [BATCHID], [PROCEDURE], [UNITPROCEDURE], [OPERATION], [PHASE], [DESCRIPT], [EVENT], [PVAL], [EU], [AREA], [PROCESSCELL], [UNIT], [PHASEMODULE], [USERID] or [USER], [UNIQUEID], [MATERIALNAME], [MATERIALID], [LOTNAME], [LABEL], [CONTAINER]

ABB 800xA Place Holders

- [TIME], [UNIQUEID], [BATCHID], [UNIT], [PROCEDURE], [UNITPROCEDURE], [OPERATION], [PHASE], [PARAMETER], [VALUE]

Value now. Value over time.

Tag Templates Functionality
Examples

- Allow for creation of PI tags, aliases, and data associated with any Event Type
- Support two extremes (and everything in-between): single events can be stored in different PI tags and multiple events can be stored in a single PI tag.

EXAMPLE TAG TEMPLATES

Single Event into Tag

```
Tag[9000301].Name=BESName:[UNIT]([PHASEMODULE]):[DESCRIPT]-[EVENT]
Tag[9000301].Descriptor=[UNIT] [PHASEMODULE] [DESCRIPT]-[EVENT]
Tag[9000301].EngUnits=[EU]
Tag[9000301].Value=[PVAL]
Tag[9000301].Type=string
Tag[9000301].UnitAlias=\\Phases\[PHASEMODULE]\[DESCRIPT] - [EVENT]
Tag[9000301].PhaseAlias=[EVENT]|[DESCRIPT]
Tag[9000301].Trigger=[EVENT,value="Recipe Value"]
Tag[9000301].Trigger=[EVENT,value="Report"]
Tag[9000301].Trigger=[EVENT,value="Owner Change"]
Tag[9000301].Trigger=[EVENT,value="Prompt"]
Tag[9000301].Translate=FALSE
```

```
TAG = BESName:RE1560(CHARGE_DIW):ACT_CHARGE_AMOUNT-Report
 03/03/2009 14:24:03.000   2414.5
```

Multiple Events into Tag

```
Tag[9000202].Name=BESName:[UNIT].Event.Multi.Recipe_Report_OwnerChange_Prompt
Tag[9000202].Descriptor=[UNIT].Event.Multi.Recipe_Report_OwnerChange_Prompt
Tag[9000202].EngUnits=PHASEMODULE.EVENT.DESCRIPT: PVAL EU
Tag[9000202].Value=[PHASEMODULE].[EVENT].[DESCRIPT]: [PVAL] [EU]
Tag[9000202].Type=string
Tag[9000202].UnitAlias=Event.Multi.Recipe_Report_OwnerChange_Prompt
Tag[9000202].Trigger=[EVENT,value="Recipe Value"]
Tag[9000202].Trigger=[EVENT,value="Report"]
Tag[9000202].Trigger=[EVENT,value="Owner Change"]
Tag[9000202].Trigger=[EVENT,value="Prompt"]
Tag[9000202].Trigger=[EVENT,value="Prompt Response"
Tag[9000202].Translate=FALSE
```

```
TAG = BESName:RE1560.Event.Multi.Recipe_Report_OwnerChange_Prompt
 03/03/2009 14:24:01.000   CHARGE_DIW.Recipe Value.CPP_HIGH_LIMIT: 2535 kg
 03/03/2009 14:24:01.000   CHARGE_DIW.Recipe Value.SP_CHARGE_MATERIAL: PW100
 03/03/2009 14:24:01.000   CHARGE_DIW.Recipe Value.SP_CHARGE_AMOUNT: 2480 kg
 03/03/2009 14:24:03.000   CHARGE_DIW.Report.ACT_CHARGE_AMOUNT: 2414.5 kg
```

Same event written to two tags, in two different formats

Value now. Value over time.

Tag Templates Functionality

Advanced Parsing Parameters

- OR conditions for Placeholders can be defined by listing multiple .Trigger attributes for the tag template.
- AND conditions for Placeholders can be defined by specifying multiple Placeholders within a single .Trigger attribute.

```
Tag[9000205].Name=BESName:[UNIT].Event.Multi.AllBatchEvents.Unit.Critical
Tag[9000205].Descriptor=[UNIT] Critical Unit Batch Events
Tag[9000205].EngUnits=BATCHID_UNIQUEID_PROCEDURE_UNITPROCEDURE_OPERATION_PHASE_DESCRIPT_EVENT_PVAL_EU_UNIT_PHASEMODULE_USERID
Tag[9000205].Value=[BATCHID]|[UNIQUEID]|[PROCEDURE]\[UNITPROCEDURE]\[OPERATION]\[PHASE]|[DESCRIPT]|[EVENT]|[PVAL]|[EU]|[UNIT]|[PHASEMODULE]|[USERID]|
Tag[9000205].Type=string
Tag[9000205].UnitAlias=Event.Multi.AllBatchEvents.Unit.Critical
Tag[9000205].Trigger=[EVENT,value="Active Step Change Commencing"]
Tag[9000205].Trigger=[EVENT,value="Comment"]
Tag[9000205].Trigger=[EVENT,value="Recipe Data Changed"]
Tag[9000205].Trigger=[EVENT,value="Recipe Value Change"]
Tag[9000205].Trigger=[EVENT,value="State Change"] [PVAL, value="HELD"]
Tag[9000205].Trigger=[EVENT,value="State Change"] [PVAL, value="RESTARTING"]
Tag[9000205].Trigger=[EVENT,value="State Change"] [PVAL, value="ABORTED"]
Tag[9000205].Trigger=[EVENT,value="System Message"] [DESCRIPT, value="Phase Logic Failure*"]
Tag[9000205].Translate=FALSE
```

Tag Templates Functionality
PIEVENTs

- The Batch Interface is capable of providing its activity on PI Batch database by generating its own **PIEvents**
  - These events are based on the triggering batch event logic the interface uses against each source system to trigger PI Batches, PIUnitBatches, PISubBatches (Operations, Phases, Phase States, Phase Steps).
  - This functionality allows customers to configure Tag Templates based on these PIEvents to write batch triggering data to PI tags (the interface is already creating PI Batch records in the PI Batch Database).

PIEVENT Example 1: PI Batch Active Tag

Tag[11].Name=BESName:PIEvent.Batch.Active
Tag[11].Value=BATCH START: [BATCHID] |Prod: [PRODUCT] |Rec: [PROCEDURE]
Tag[11].Trigger=[EVENT,value="PIEVENT"] [DESCRIPT, value="BATCH"] [PVAL,value="START"]
//// SAME TAG
Tag[12].Name=BESName:PIEvent.Batch.Active
Tag[12].Value=BATCH END: [BATCHID] |Prod: [PRODUCT] |Rec: [PROCEDURE]
Tag[12].Trigger=[EVENT,value="PIEVENT"] [DESCRIPT, value="BATCH"] [PVAL,value="END"]

PIEVENT Example 2: PI Unit Batch Active Tag

Tag[21].Name=BESName:[UNIT].PIEvent.UnitBatch.Active
Tag[21].Value=1
Tag[21].Type=integer
Tag[21].UnitAlias=PIEvent.UnitBatch.Active
Tag[21].Trigger=[EVENT,value="PIEVENT"] [DESCRIPT, value="UNITBATCH"] [PVAL,value="START"]
//// SAME TAG
Tag[22].Name=BESName:[UNIT].PIEvent.UnitBatch.Active
Tag[22].Value=0
Tag[22].Type=integer
Tag[22].UnitAlias=PIEvent.UnitBatch.Active
Tag[22].Trigger=[EVENT,value="PIEVENT"] [DESCRIPT, value="UNITBATCH"] [PVAL,value="END"]

PIEVENT Example 3: PI Unit Batch BatchID Tag

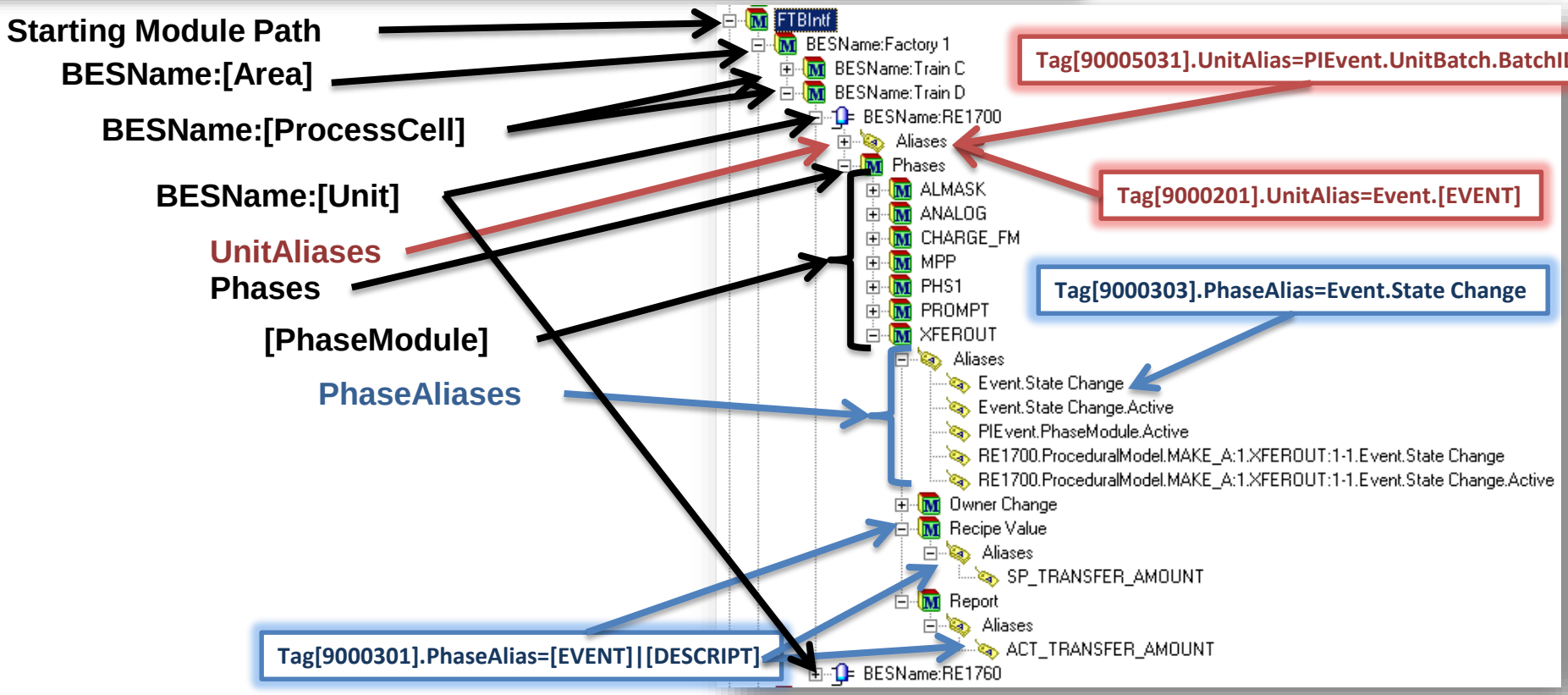
Tag[31].Name=BESName:[UNIT].PIEvent.UnitBatch.BatchID
Tag[31].Value=[BATCHID]
Tag[31].UnitAlias=PIEvent.UnitBatch.BatchID
Tag[31].Trigger=[EVENT,value="PIEVENT"] [DESCRIPT, value="UNITBATCH"] [PVAL,value="START"]
//// SAME TAG
Tag[32].Name=BESName:[UNIT].PIEvent.UnitBatch.BatchID
Tag[32].Value=Inactive
Tag[32].UnitAlias=PIEvent.UnitBatch.BatchID
Tag[32].Trigger=[EVENT,value="PIEVENT"] [DESCRIPT, value="UNITBATCH"] [PVAL,value="END"]

Tag Templates Functionality

PI Module Creation

- The Batch Interface performs automated module and unit creation within the PI Module DB on the PI Server.
  - An optional Starting Module Path (/smp command line parameter) can be defined in the interface .bat startup file. Example: /smp=FTBIntf

[GENERAL]
Equipment=BESName:[Area]\BESName:[ProcessCell]\BESName:[Unit]\Phases\[PhaseModule]

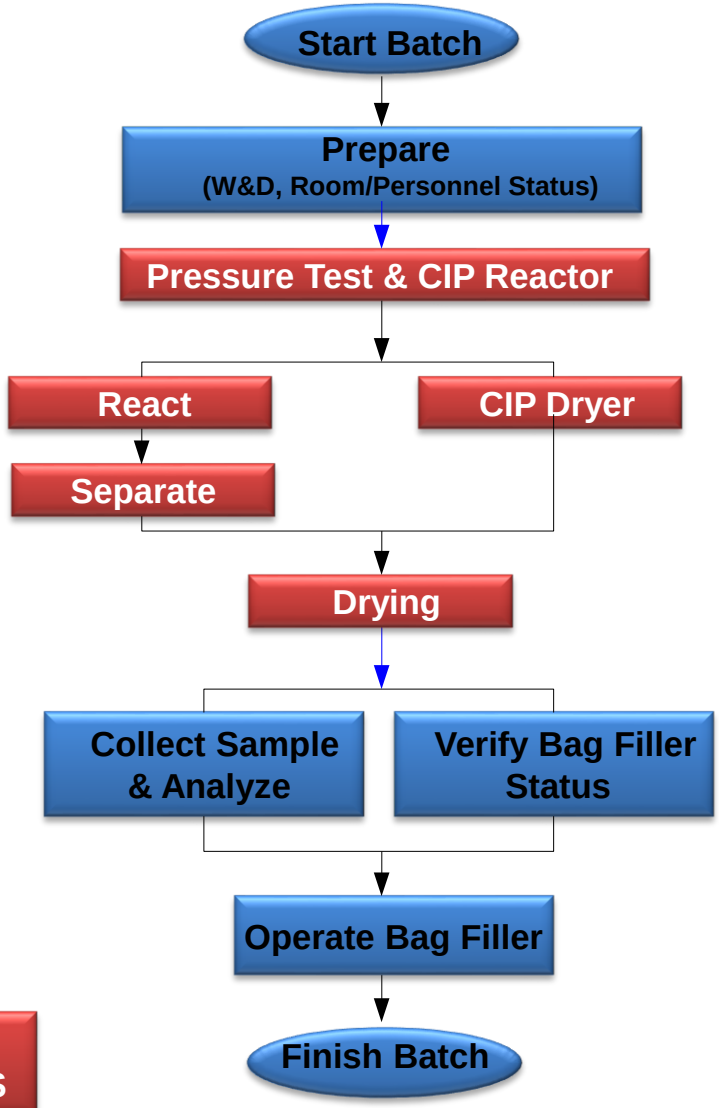


Value now. Value over time.

Batch Interface Framework Functionality

Merging Multiple Source Batches into a Single PIBatch

- The Batch Interface is capable of merging multiple source batches into a single PIBatch
- Example Use Cases:
  - Customers that run smaller batch recipes in a flexible manufacturing environment but want to report on the data as one larger batch
  - Customers that run a Batch Execution System (example: DeltaV) and a Manufacturing Execution System (example: Syncade) and want to merge automated steps with manual electronic work instructions.
- Configuration (interface .bat startup file):
  - **/merge** – Source batches with the same BatchID are merged into one PIBatch.
  - **/cachetime=X** – Specifies the duration in days (can also be a fraction of the day) for which the interface keeps the closed batches in the local cache memory. [Default = 1 day]
  - **/bidm=<string>** - [OPTIONAL] **B**atch**ID** **M**ask: allows the interface to use a substring of the source batch BatchID as the BatchID for the PIBatch. Wildcards are supported.



Manual Activities in Manufacturing Engine MES

Automated Activities in Batch Execution Engine DCS

Value now. Value over time.

Batch Interface Framework Product Development Roadmap

- Planned Batch Interface Framework Interfaces
  - PI Batch Generator (PIBaGen)
  - Siemens PCS7 Batch Interface
  - Wonderware InBatch Interface
  - Werum PAS-X Batch Interface
  - …
- N-Way Buffering for Batch Interfaces (after PI 3.4.380 SP1/3.5 release)
  - Will provide ‘fanning’ of batch data to multiple PI Servers in a HA collective
  - Ability to create the same PI Batch GUID on both PI Servers
  - Will enable ‘batch HA’ for 90% of customer use cases
- Transition to Event Frames
  - Swap out back-end writes to EF database instead of PI Batch DB
  - Enable auto-configuration of AF database instead of Module DB
  - Provide enhanced tag template functionality for writing to AF/EF attributes

More Information

- Monthly OSIsoft Technical Support News Letter
- OSIsoft Technical Support Website
  - http://Techsupport.OSIsoft.com
  - Release announcements
  - Support Bulletins or Knowledge Base (KB) Articles for an interface
- Interfaces Product Manager – Chris Coen
  - ProductManagers@osisoft.com

OSIsoft has developed and will continue to develop the interfaces that provide the critical data required for you to run your business

Thank you

777 Davis St., Suite 250 San Leandro, CA 94577